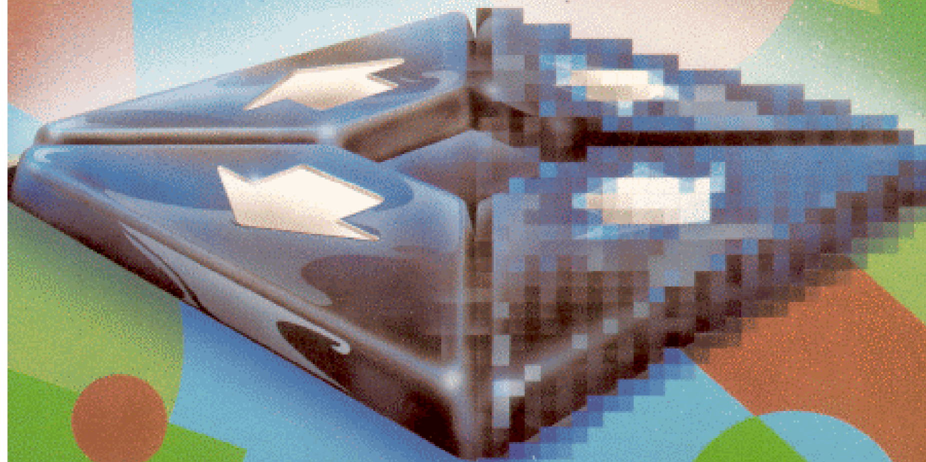
GRAFISCHE EXPERIMENTEN VOOR MSX-COMPUTERS

MET BEELDBEWERKINGSPROJECT

W.H.M. van Dreumel



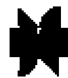
Kluwer

W.H.M. van Dreumel

Grafische experimenten voor MSX-computers

met beeldbewerkingsproject

Deel 2
Het Beeldbewerkingsproject



Kluwer Technische Boeken B.V.
Deventer- Antwerpen

Scanned, ocr'ed and converted to PDF by HansO, 2002
Programma's zijn niet gegarandeerd foutvrij!

Inhoud

Deel 2 Het Beeldbewerkingspoject

18 Inleiding deel 2 5
19 Het elektronicaproject 6
Afbeelding 19-2 Een algemene opzet voor het meten van de helderheid met ..7
20 De fotogevoelige opnemer 13
21 De koppeling met de computer 18
22 Een schakeling voor gevorderde elektronici..24
23 Printeraanpassing 25
24 Het werken met de digitizer...30
25 Beeldopslag 33
26 Beeldbewerking..39
27 Geometrische beeldbewerking..44
28 Uitvoer op de printer..51
29 Iets over het geheugen 55

Deel 2

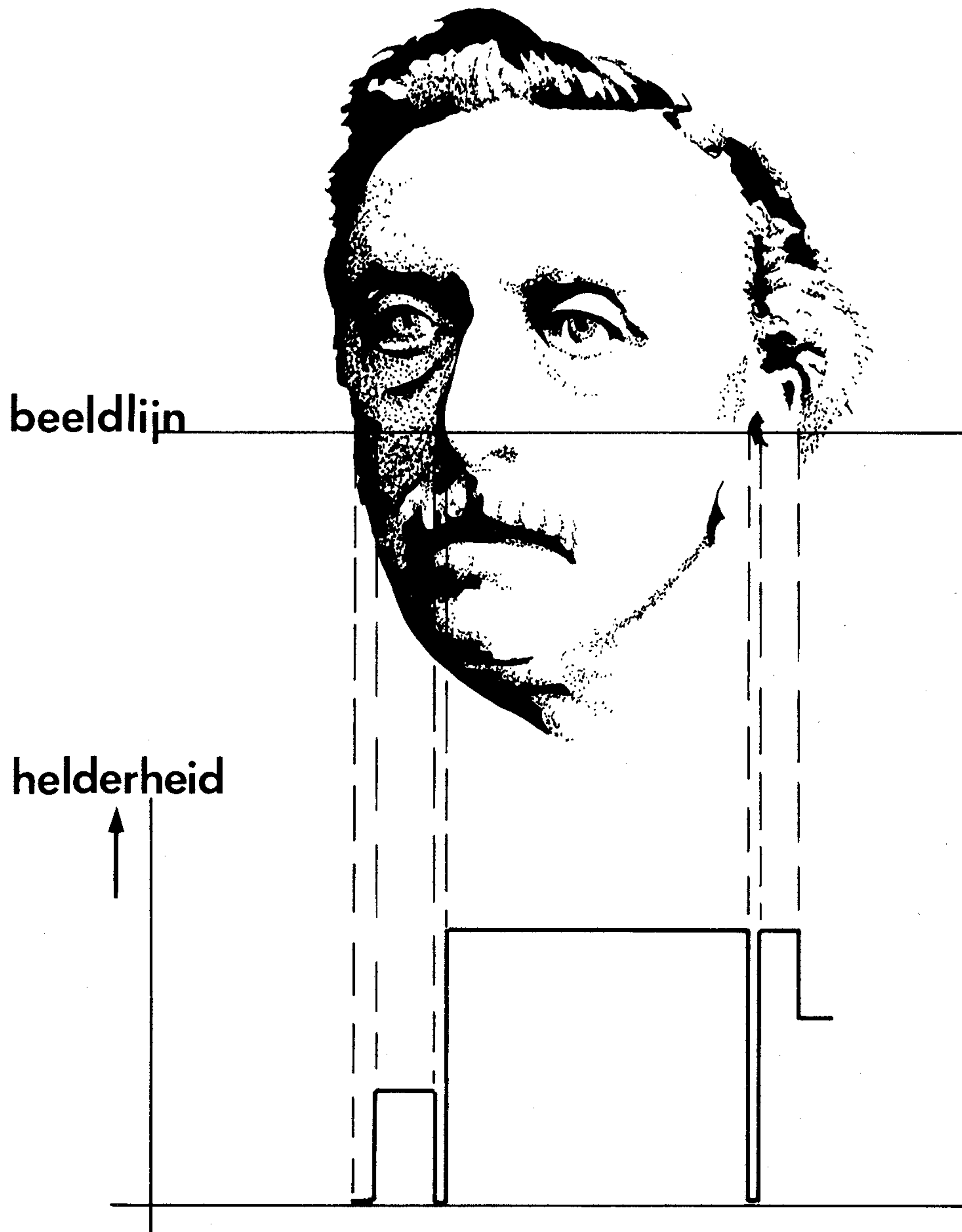
Het Beeldbewerkingsproject

18 Inleiding deel 2

In dit deel gaan we samen op avontuur in de wondere wereld van 'advanced computer graphics'. We zullen kennismaken met veel aspecten van elektronische beeldvorming en de theorie ook echt in praktijk brengen. Hoewel de geschetste technieken op alle soorten computerbeelden toepasbaar zijn, geeft het bewerken van zelf-opgenomen beelden toch wel de grootste bevrediging. We beginnen dan ook met een eenvoudig elektronicaproject dat door iedereen uit te voeren is. In dit project zullen we de bouw beschrijven van een optische leeskop waarmee we zelf foto's, illustraties en andere afbeeldingen om kunnen zetten in een computerbeeld. Om het geheel ook voor de beginner geschikt te maken, zullen we moeilijke interfaceproblemen omzeilen door gebruik te maken van de joystick-ingang. Op deze wijze gaan we het risico dat schade wordt aangebracht aan het inwendige van de computer uit de weg. Hoewel het project geschikt is voor alle computers met een joystick-ingang, zijn de begeleidende programma's geschreven in MSX-BASIC. Het gebruik van speciale MSX-faciliteiten is echter zo veel mogelijk vermeden, zodat omzetting naar andere BASIC-dialecten in de meeste gevallen geen grote problemen zal opleveren.
Lees dit deel van het boek in de onmiddellijke nabijheid van de computer. De vele kleine programma's kunnen dan tijdens het lezen al worden uitgeprobeerd, zodat de werking ervan duidelijk is bij de behandeling van het complete beeldverwerkingsprogramma in het laatste hoofdstuk. Met dit programma kunnen beelden opgenomen, weergegeven, bewerkt en afgedrukt worden.

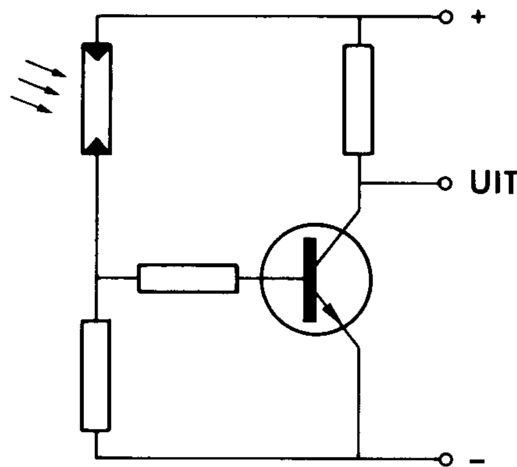
19 Het elektronicaproject

Dat de MSX-computer ongekende grafische mogelijkheden heeft, wordt door niemand ontkend.



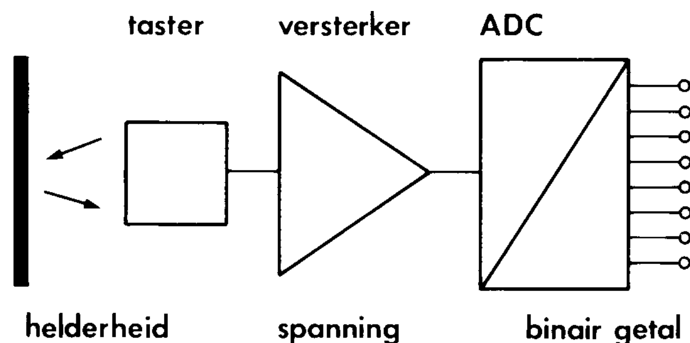
Afbeelding 19-1 De helderheid als functie van de plaats op een beeldlijn.

Toch kan het vullen van het grafisch scherm met een gedetailleerde afbeelding nog aardig wat problemen opleveren. Hoe zit dat dan met die fraaie computerbeelden waarop de wereld gedigitaliseerd weergegeven is? Of concreter gesteld: Hoe krijgen we een afbeelding vanaf papier rechtstreeks op het scherm ?
We duiken even in de TV-techniek. Dat het TV-beeld is opgebouwd uit lijnen mag als bekend worden verondersteld. Afbeelding 19-1 geeft zo'n lijn weer. De helderheid langs de lijn varieert: alle grijstinten tussen zwart en wit kunnen worden afgebeeld. Het meten van de helderheid langs zo'n lijn is met wat elektronica nog wel te realiseren. Afbeelding 19-2 geeft hier een algemene opzet voor.



Afbeelding 19-2 Een algemene opzet voor het meten van de helderheid met behulp van een lichtgevoelige weerstand.

Als we de taster langs een lijn van een afbeelding bewegen, verschijnt er op de uitgang een signaal waarvan de hoogte afhankelijk is van de grijswaarde van de afbeeldmg. Een analoog signaal dus. Met een analoog/digitaalconverter kan dit signaal gedigitaliseerd worden, zodat het door de computer te verwerken is (zie afbeelding 19-3).



Afbeelding 19-3 Het omzetten van de helderheid in een getal dat door de computer te lezen is.

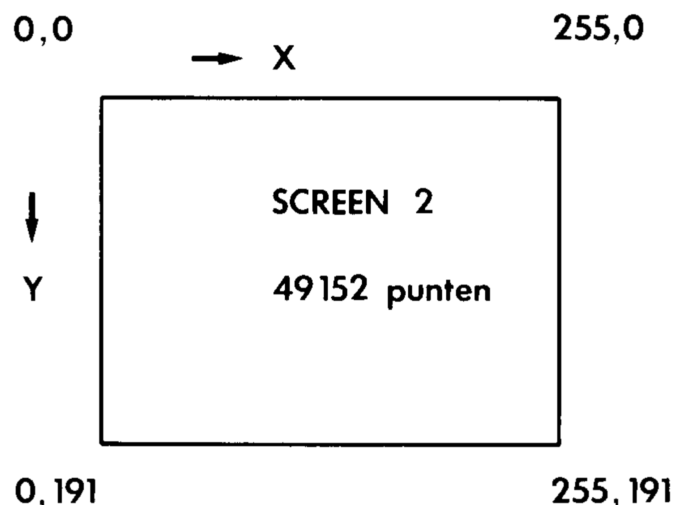
Wat ons te doen staat is:

- Het bouwen van een lichtopnemer die de intensiteit van het door de beeldpunten op papier gereflecteerde licht meet en omzet in een elektrisch signaal.
- Beslissen op welke wijze het signaal aan de computer wordt aangeboden.
- De bouw van een analoog/digitaalconverter (ADC) gebaseerd op de keuze uit het vorige punt.
- Het verzinnen van een aftastmechanisme om zodoende een afbeelding lijn voor lijn in het computergeheugen te kunnen opslaan.

Voor we met ongeremd enthousiasme met het ontwerpen van de diverse componenten beginnen, lijkt het niet onverstandig ons eisenpakket wat beter te omschrijven.
De activiteiten van de MSX-computer spelen zich af op vier mogelijke scherm-soorten, waarbinnen zich weer een aantal varianten aftekenen ten aanzien van sprite-af metingen.

SCREEN 0	Een tekstscherm van 24 regels met elk 40 tekens.
SCREEN 1	Een tekstscherm van 24 regels met 32 iets bredere tekens.
SCREEN 2	Een scherm voor grafisch gebruik met een hoog oplossend vermogen van 256 bij 192 beeldpunten.
SCREEN 3	Een grafisch beeldscherm met een laag oplossend vermogen van 64 bij 48 beeldpunten.

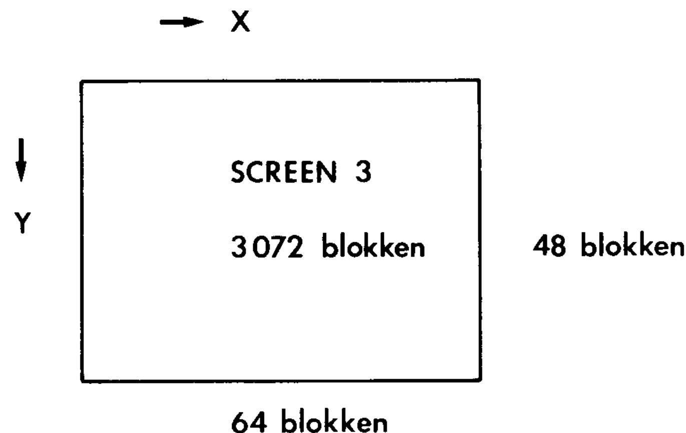
Het zal duidelijk zijn dat we voor beeldvorming op de schermsoort 2 of 3 aangewezen zijn. Een nadere beschouwing van deze schermen is dan ook wenselijk. Afbeelding 19-4 geeft schermsoort 2 weer.



Afbeelding 19-4 De opbouw van scherm 2.

Een horizontale beeldlijn is opgebouwd uit 256 beeldpunten. Helemaal links ligt beeldpunt O, terwijl beeldpunt 255 zich geheel aan de rechterzijde bevindt. Elk

beeldpunt laat zich met de PSET-opdracht aansturen. Een horizontale lijn op het hogeresolutiescherm is opgedeeld in groepjes van acht beeldpunten. Binnen zo'n groepje van acht mag slechts één enkele kleur voorkomen (deze beperking geldt niet voor punten langs verticale lijnen). Hoewel deze wet de grafische mogelijkheden van de MSX beperkt, betekent het voor ons project een aanzienlijke vereenvoudiging. Door de beperking is het minder nuttig een optische lezer te ontwikkelen waarmee elk beeldpunt afzonderlijk kan worden uitgelezen als we die punten toch niet individueel kunnen beïnvloeden. We mogen op een minder fijne schaal werken. Dat sluit prachtig aan op de mogelijkheden van het andere grafische scherm: scherm 3 (zie afbeelding 19-5).



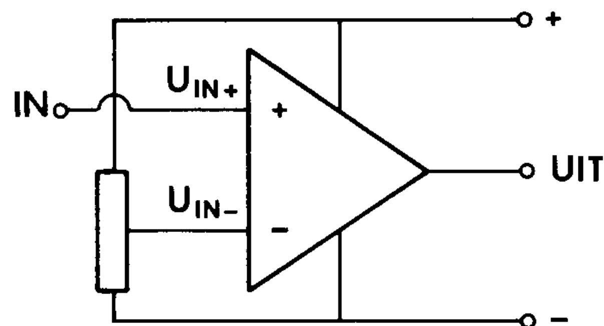
Afbeelding 19-5 De opbouw van scherm 3.

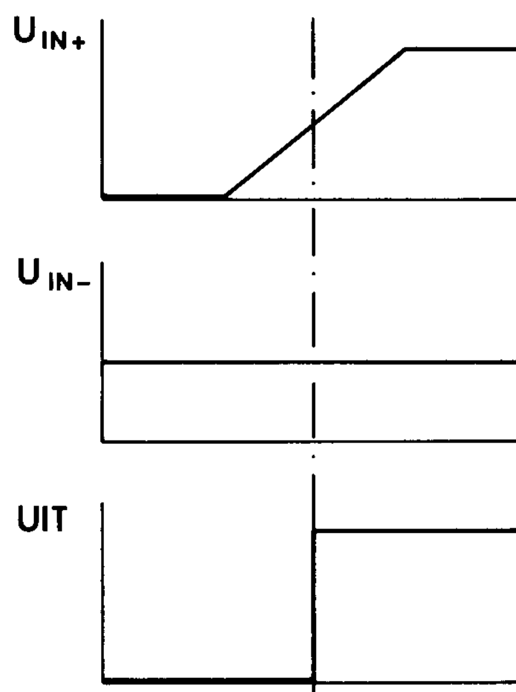
Scherm 3 wordt ook wel het multicolor-scherm genoemd, omdat de beperking van het hogeresolutiescherm hier niet aanwezig is. Elk beeldpunt kan elke mogelijke kleur aannemen. Een beeldpunt bestaat echter uit een vlakje van vier bij vier hogeresolutiepunten.

Hoewel de beeldpunten dus aanzienlijk groter zijn, is op een horizontale lijn het kleuroplossend vermogen tweemaal zo groot. Terwijl op het hogeresolutiescherm horizontale lijntjes van acht punten eenzelfde kleur moesten hebben, kunnen we op het multicolor-scherm lijntjes van vier beeldpunten een kleur toekennen.

De optische lezer, die we nodig hebben om beelden op papier om te zetten in elektrische signalen, moet in staat zijn de gemiddelde grijswaarde te meten van een vlakje dat ongeveer overeenkomt met de afmeting van een lageresolutiebeeldpunt: een vierkantje van ongeveer een halve bij een halve centimeter.

Ten aanzien van de keuze van de benodigde analoog/digitaal-omzetter gelden ook beperkende voorwaarden. Het eenvoudigst is een omzetter die slechts twee niveaus kent. Twee niveaus worden namelijk al gedetecteerd door de eenvoudige vergelijkerschakeling uit afbeelding 19-6.





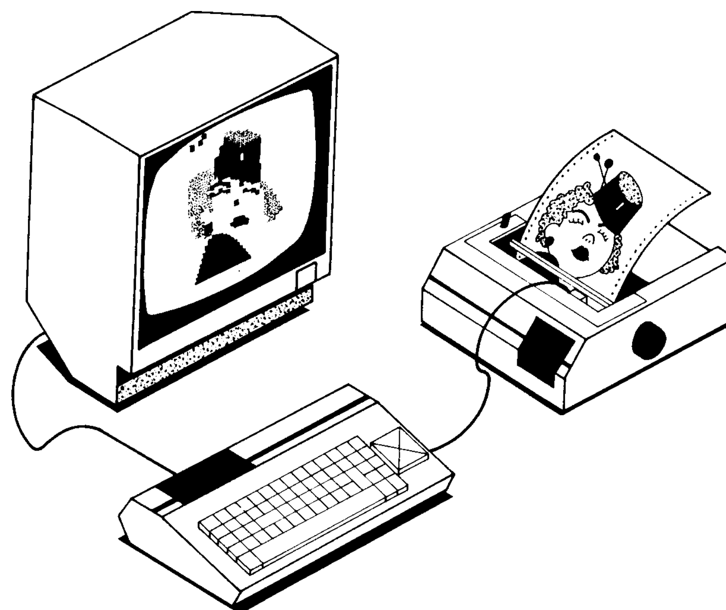
Afbeelding 19-6 Een eenvoudige vergelijkerschakeling.

Als de ingangsspanning boven het met de potentiometer ingestelde niveau ligt, is de uitgang 'hoog'. Ligt de ingangsspanning eronder, dan is de uitgang 'laag'. Met deze schakeling kunnen we een beeld in een zwart/witplaatje omzetten. Maar dan ook letterlijk zwart/wit, dus zonder tussenliggende grijstinten. Hoewel met de geschetste opzet best aardige dingen te doen zijn, is de beperking wel erg groot. Er bestaat ook een ander uiterste. Stel dat we kiezen voor een ADC met een acht-bits-uitgang. Een voor de hand liggende gedachte omdat de in de MSX toegepaste Z80-microprocessor met woorden van acht bits werkt. We kunnen dan 2 tot de macht 8 ofte wel 256 kleurtinten vastleggen. Maar hoe doen we dat op een computer met 'slechts' 16 kleurmogelijkheden? Als we een zwart/wit-TV aansluiten en we gaan ervan uit dat elke op de MSX beschikbare kleur een goed te onderscheiden grijswaarde oplevert, dan komen we al niet verder dan 15

grijswaarden (transparant telt niet mee). Bovendien willen we het gebruik van kleur niet opgeven, omdat het nu juist de charme aan de computer verleent.
De volgende tabel geeft de kleurmogelijkheden van MSX-systemen, iets anders gerangschikt dan in de meeste handboeken:

Kleur	Nummer	Kleur	Nummer
Lichtrood	9	Lichtgroen	3
Rood	8	groen	2
Donkerrood	6	donkergroen	12
Cyaan	7	helderwit	15
Lichtblauw	5	wit	14
Donkerblauw	4	Lichtgeel	11
Paars	13	geel	10
transparant	0	zwart	1

De kleuren rood en groen zijn met drie helderheden vertegenwoordigd. Eigenlijk geven alleen deze kleurschakeringen een harmonische contrastcombinatie. Een ADC met drie niveaus lijkt voor een niet al te ingewikkeld beeldverwerkingssysteem dan ook aantrekkelijk. De consequentie van het toepassen van een ADC met drie niveaus is duidelijk. Het beeld verliest aan nuancering en wordt opgedeeld in drie helderheidsgraden. Alle grijswaarden binnen een bepaald helderheidsgebied worden bij elkaar geveegd. In de fotografie staat dit verschijnsel bekend als toonscheiding. Door het indringende effect wordt toonscheiding veel gebruikt als kunstzinnige uiting. Ook in de reclamefotografie komen we toonscheiding vaak tegen. Dat met slechts drie 'tonen' toch aantrekkelijke resultaten te realiseren zijn, laten de afbeeldingen in dit boekzien.

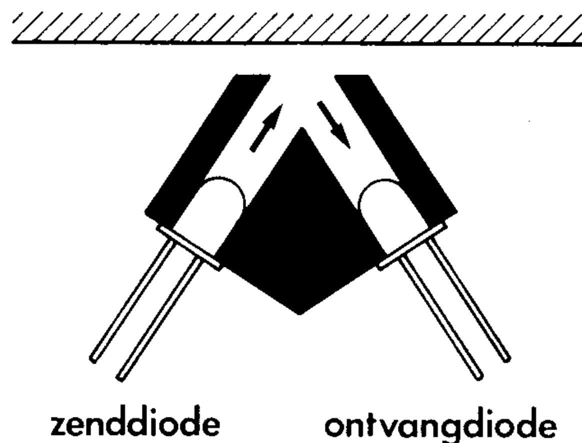


Afbeelding 19-7 Het beeldverwerkingssysteem.

Als laatste punt dient de beeldopnemer zich aan. We willen lijnen aftasten. We kunnen de beeldopnemer langs een liniaal bewegen en om de paar millimeter een helderheidsmeting doen. Vervolgens verschuiven we de liniaal iets naar onderen en tasten de volgende lijn af. Omslachtig en onpraktisch, nietwaar? Vrijwel elke computerbezitter beschikt over een printer. U raadt het al: een printkop beschrijft een horizontale lijn en keert daarna terug naar het begin van de regel. Vervolgens wordt het papier een regel opgeschoven en een nieuwe lijn wordt afgewerkt. Als we op de bewegende printkop een lichtgevoelig opnemertje monteren en we de op te nemen afbeelding op het printerpapier plakken, krijgen we een prachtige beeld-scanner.
In de volgende hoofdstukken komen de opnemer, de koppeling met de computer, de analoog/digitaal-omzetter en de ombouw van de printer aan de orde. Dit laatste overigens zonder de printer ingrijpend te wijzigen. Afbeelding 19-7 geeft een indruk van het hele beeldverwerkingssysteem.

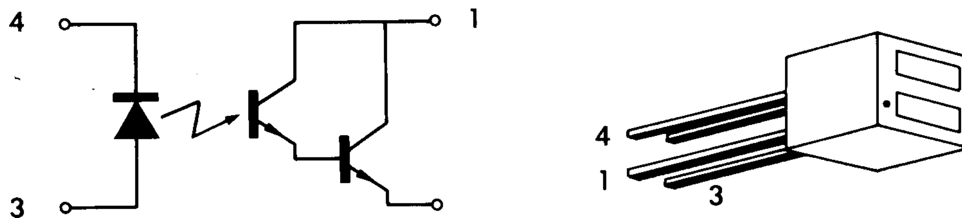
20 De fotogevoelige opnemer

Zoals we in het vorige hoofdstuk hebben afgeleid, moet het lichtgevoelige opneemelement werkzaam zijn op een oppervlakte van ongeveer een kwart vierkante centimeter. Er zijn uiteraard meer eisen te verzinnen. De opnemer moet compact en licht zijn. We kunnen van een printer niet eisen dat hij een grote print vol elektronicacomponenten meevoert.
Globaal gesproken heeft een printer één seconde nodig om een volledige lijn af te werken. Op een horizontale lijn van screen 3 liggen 64 beeldpunten. De opnemer moet dus in staat zijn per seconde 64 lichtveranderingen te registreren. Laten we er gemakshalve van uitgaan dat de opnemer een frequentiebe-reik van honderd hertz moet hebben. Hierdoor komt een lichtgevoelige weerstand (LDR) niet voor onze toepassing in aanmerking. Een fototransistor of -diode is echter bij uitstek geschikt.
De opnemer moet zo ongevoelig mogelijk zijn voor omgevingslicht. Een fotodiode in het infraroodgebied zou al heel wat problemen met daglichtafscherming oplossen. Het spreekt welhaast vanzelf dat we als lichtbron in dit geval een infrarood-LED gebruiken. Afbeelding 20-1 laat een opstelling zien voor een zend- en ontvangdiode. De ontvangdiode zet het door de afbeelding gereflecteerde licht om in een elektrische stroom.



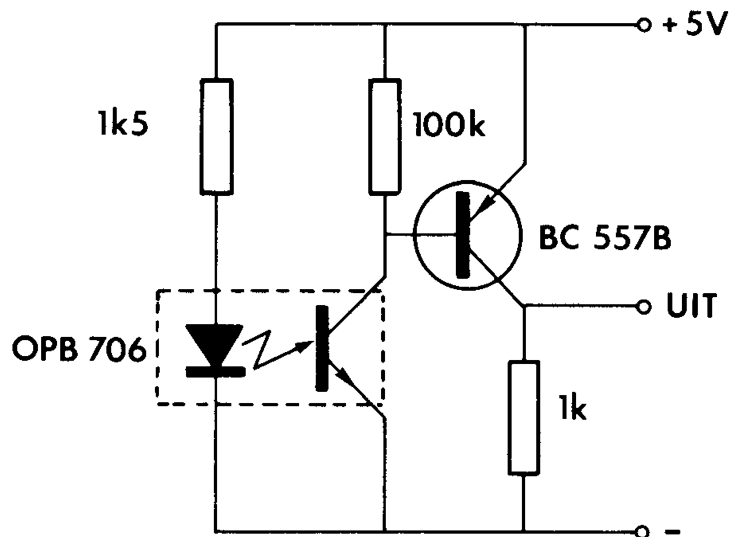
Afbeelding 20-1 De zend- en ontvangdiode, beide werkzaam in het infraroodgebied.

Zend- en ontvangdioden voor infrarood licht zijn ruimschoots verkrijgbaar. Met enige handigheid kan het afgebeelde opnemertje zelf worden vervaardigd.
Nog fraaier is de toepassing van een kant en klaar element, zoals de OPB-706, de CNY-70 of één van de vele varianten. Deze opnemer is bedoeld om diffuus reflecterende oppervlakken af te tasten en voldoet dan ook bij uitstek aan het door ons gestelde doel.



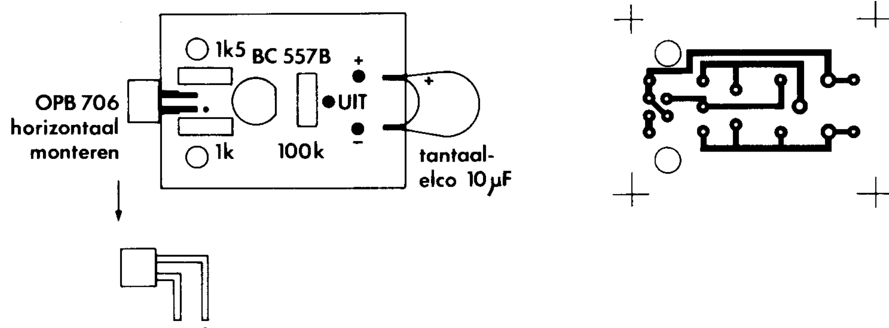
Afbeelding 20-2 De OPB-706: een combinatie van een infrarood zenddiode en een infrarood ontvangdiode.

Het gebruik van een fotodiode of een fototransistor heeft wel tot gevolg dat we flink wat signaalversterking in moeten bouwen. En dat zal dan met een minimum aan onderdelen moeten gebeuren, omdat we maar een klein printje in de printer kunnen bouwen. Het schema uit afbeelding 20-3 helpt ons al een aardig eind op weg.



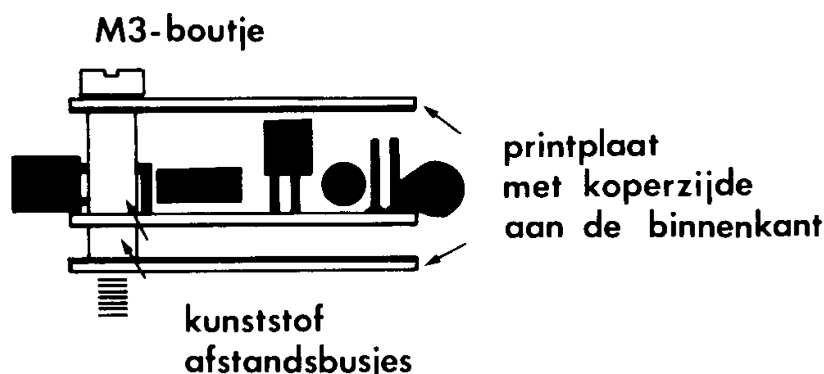
Afbeelding 20-3 Het schema van de lichtopnemer.

De opnemer, een PNP-transistor, drie weerstanden en een tantaal-elco. Dat is, behalve het printje, alles wat we nodig hebben voor de ontvanger. Voor het zendgedeelte komt daar nog een weerstand bij.
Het printje heeft ook al niet veel om het lijf. De lezer die in staat is zelf printen te vervaardigen, zal met dit miniprintje geen enkel probleem hebben. De sporen kunnen op het printmateriaal worden getekend met een etsbestendige pen. Ook kunnen etsbestendige afwrijfsymbolen worden gebruikt; die zijn tegenwoordig in vrijwel elke elektronicazaak te krijgen. Het kan ook zonder geëtste print. Een stukje gaatjesprint geeft voldoende basisvoor het stabiel verbinden van de onderdelen.
Door de aftaster (sensor) een plaatsje aan de rand van de printte geven, kunnen we voor elke printer een optimale stand realiseren. Zo kan de sensor ge-



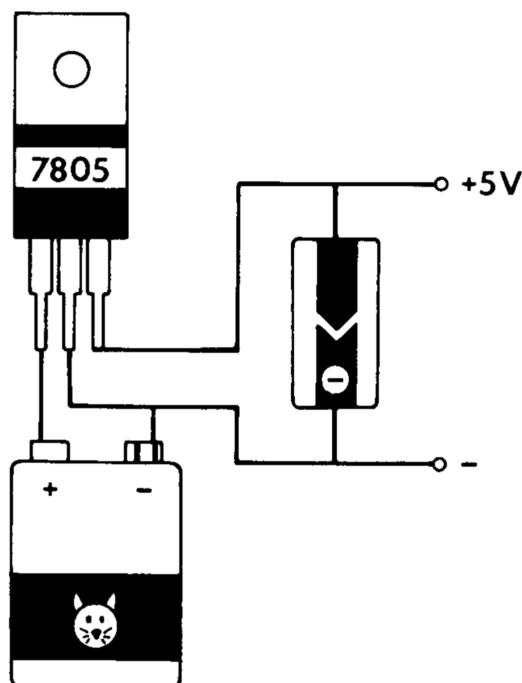
Afbeelding 20-4 De print-layout en de componentenopstelling van de lichtopnemer.

woon op de print worden gesoldeerd, waarbij de lengte van de poten aan de omstandigheden kan worden aangepast. Laat ze niet langer dan strikt noodzakelijk, omdat het draadeinde tussen de ontvangdiode en de versterkeringang brom kan oppikken. In de meeste gevallen is het handig als de sensor evenwijdig aan het printoppervlak wordt gemonteerd.
De print is zó klein, dat de bevestiging op de printerslede geen onoverkomelijke moeilijkheden hoeft op te leveren. In hoofdstuk 23 komen we hierop terug.
Mocht het nodig zijn dan kan de hoeveelheid LED-licht aan de omstandigheden worden aangepast door de weerstand van 1,5 kQ te veranderen. Een lagere waarde dan 270 Q is echter niet toegestaan. Dit in verband met de maximaal toegestane LED-stroom.
Detaantaal-elco zorgt vooreen extra ontkoppeling van de voedingslijnen. Om ruimte te besparen, worden de aansluitsnoeren rechtstreeks in de daarvoor bestemde printgaatjes gesoldeerd. De tantaal-elco wordt hierbij misbruikt als trekontlasting. Een plakbandjeom deaansluitdradenen de condensator voorkomt dat de aansluitingen door herhaald verbuigen, afbreken. Neem voor de aansluitdraden dun en soepel geïsoleerd montagedraad van voldoende lengte. Vlecht de draden in elkaar. Eventueel kan dun tweeaderig afgeschermd snoer (stereo-kabel) worden toegepast. In dat geval wordt een ader gebruikt.



Afbeelding 20-5 De afscherming van de lichtopnemer.

voor het signaal en een voor de voedingsspanning. De afscherming dient als massaverbinding.
Vanwege de hoge signaalversterking is het raadzaam om het printje af te schermen. Dit kan op een zeer eenvoudige manier. Twee miniprintplaatjes, aan elke zijde één, maken er een sandwich van (zie afbeelding 20-5). Ze voorkomen dat omgevingsbrom wordt opgepikt. Door middel van een draadje komt de massa-aansluiting met de print tot stand.
We gebruiken de printplaatjes tevens voor een ander doel. De afstand van de sensor tot het papier waarop de over te nemen afbeelding staat, dient redelijk constant te zijn. Immers, de hoeveelheid gereflecteerd licht dat de ontvanger bereikt, wordt beïnvloed door de afstand tot het reflecterende oppervlak. De laatste schets geeft aan dat de afschermende printplaatjes, in het geval dat de sensor evenwijdig aan de print is gemonteerd, voorkomt dat de afstand tot het papier varieert.
Twee M3-boutjes met de bijbehorende kunststof afstandsbusjes vervolmaken het geheel tot een zeer stabiele en compacte leeskop. De boutjes geven bovendien goede mogelijkheden voor de montage in de printer. Hoewel de uiteindelijke schakeling rechtstreeks vanuit de computer wordt gevoed, is het raadzaam in de testfase een aparte 5 V-voeding te gebruiken, om elk risico te vermijden. Een 5 V-spanning kan heel eenvoudig van een 9 V-batterijtje worden afgeleid. Een spanningsregelaar van het type 7805 en een elco van 470 (iFzorgen vooreen goed bruikbarespanningsbron(zieaf beelding 20-6). Vóór we de ADC onder de loep nemen, testen we de leeskop door hem op een zwart gedeelte van een krant te zetten. De uitgangsspanning dient in dat geval ongeveer één volt te bedragen. Plaatsen we de leeskop op een onbedrukt ge-



Afbeelding 21-5 Een alternatieve voeding

deelte van de krant, dan staat er vier tot vijf volt op de uitgang. Een grijs gedeelte van een foto zal een waarde van ongeveer twee volt opleveren. Deze waarden zijn slechts een indicatie en hangen af van de afstand tussen de sensor en het papier, de papiersoort en de gebruikte inktsoort. Zo absorberen sommige soorten viltstiftinkt vrijwel geen infrarood licht. Wat voor ons oog een hoog contrast oplevert, hoeft dus niet per se hetzelfde effect op te leveren in het infraroodgebied van het spectrum. De versterking kan eventueel nog wat worden aangepast door de waarde van de terugkoppelweerstand (10 Mohm) te veranderen.

21 De koppeling met de computer

Alvorens in dit hoofdstuk de analoog-naar-digitaal-omzetting uit te werken, zullen we de mogelijkheden eens op een rij zetten die de MSX-computer heeft om met de buitenwereld te communiceren. Dat zijn er nogal wat. De standaarduitvoering heeft maar liefst zes aansluitingen waar we een signaal in de computer kunnen voeren:

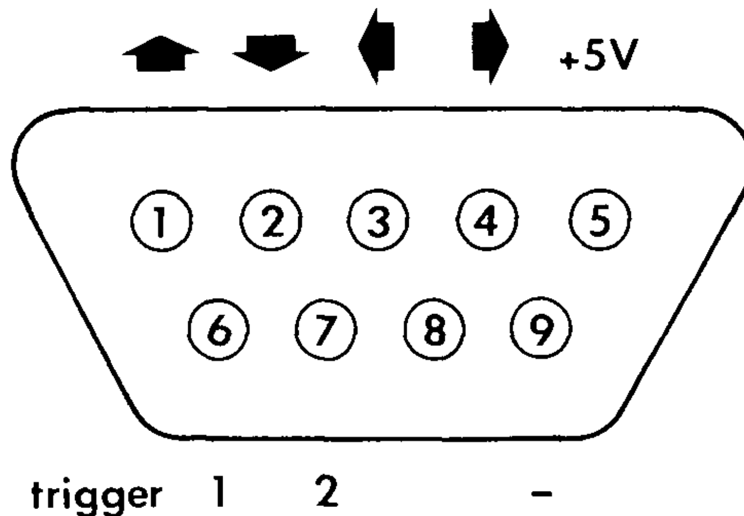
- Twee joystick-ingangen.
- Een ingang voor de cassetterecorder.
- Een gleuf waarin allerlei uitbreidingen zoals Quickdisk en ROM-packages aansluiting vinden.
- Een Centronics-aansluiting waarop een printer kan worden aangesloten.
- Een interface-aansluiting waarop vrijwel alles aanwezig is om het com-
- putergebeuren van buitenaf te regelen.

De laatste aansluiting geeft de meeste toepassingsmogelijkheden. Met enige kennis van de interne adreshuishouding, wat decodeerwerk en twee 'bus-IC's' behoort de bouw van een universele interface tot de mogelijkheden. Maar voor we naar het handboek en de soldeerbout grijpen, is enige bezinning op zijn plaats.

Laten we ons nog even goed realiseren wat ons uitgangspunt is: Een spanning omzetten naar drie discrete niveaus, en aan de hand van het geldende niveau een beeldpunt kleuren. We hebben gezien dat alleen de kleuren rood en groen elk in drie schakeringen voorkomen. Nog zonder te kiezen voor een bepaalde koppeling kan een programma er als volgt uitzien:

```
10  REM beeldpunt
20  SCREEN 2:COLOR 15,4,1
30  OPEN "grp:" FOR OUTPUT AS #1
40  PRINT #1,"[redacted]";
50  PRINT #1,"Positie x,y";
60  INPUT X,Y
70  INPUT NIVEAU
80  IF NIVEAU=1 THEN C=9:REM donkerrood
90  IF NIVEAU=2 THEN C=8:REM rood
100 IF NIVEAU=3 THEN C=6:REM lichtrood
110 PSET (X,Y),C
120 GOTO 120
```

Het behoeft geen betoog dat het zinloos is om het hier afgedrukte programma ook daadwerkelijk in te typen. De regels 70 t/m 100 laten echter duidelijk zien dat het binnenhalen van de gewenste gegevens heel goed kan via één van de joystick-ingangen. Daar hebben we immers de beschikking over meer dan voldoende onafhankelijke kanalen, namelijk de kanalen die de richtingen omhoog, omlaag, naar links en naar rechts representeren. Bovendien kunnen ze diagonale bewegingen detecteren en er zijn ook nog twee 'vuurknop'aansluitingen (zie afbeelding 21-1).



Afbeelding 21-1 De joystick-ingang van een MSX-computer.

Herschreven voor de joystick-ingang nummer 1, ziet het programma er als volgt uit:

```
10  REM joystick
20  SCREEN 2:COLOR 15,4,1
30  OPEN "grp:" FOR OUTPUT AS #1
40  PRINT #1,"[redacted]";
50  PRINT #1,"Positie x,y";
60  INPUT X,Y
70  A=STICK(1)
80  IF A=3 THEN C=9:REM donkerrood
90  IF A=5 THEN C=8:REM rood
100 IF A=7 THEN C=6:REM lichtrood
110 PSET (X,Y),C
120 GOTO 120
```

De analoog-naar-digitaal-omzetter die het werk voor ons moet gaan doen, wordt aangesloten op de joystick-ingangen rechts, omlaag en links. Om dit te illustreren laat het volgende programma een beeldpunt langs horizontale lijnen over het scherm lopen. Hierbij wordt de kleur met de cursorbesturing op het toetsenbord gekozen.

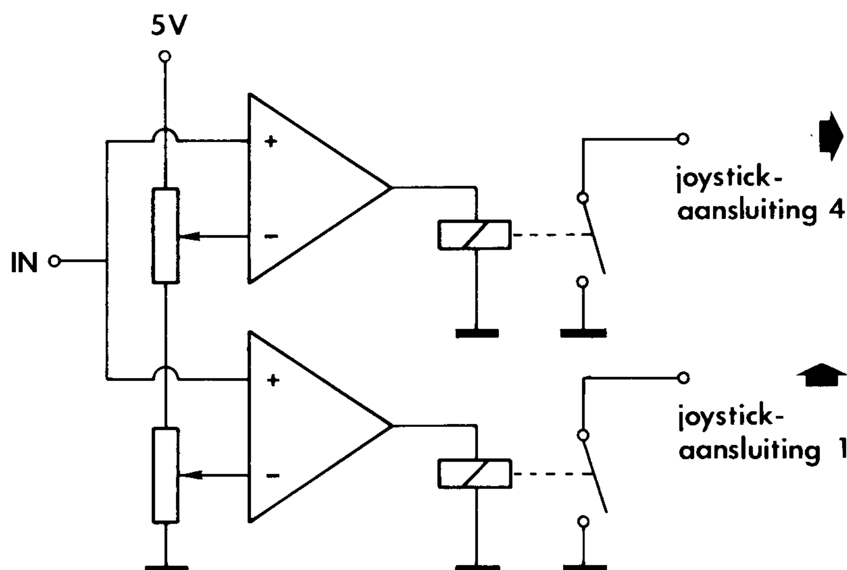
```
10 REM pixels in rood
20 SCREEN 3:COLOR 1,1,1:CLS
30 FOR Y=0 TO 192 STEP 4
40 FOR X=0 TO 255 STEP 4
50 A=STICK(0)
60 IF A=3 THEN C=9
70 IF A=5 THEN C=8
80 IF A=7 THEN C=6
90 IF A<>3 AND A<>5 AND A<>7 THEN 110
100 PSET (X,Y),C
110 NEXT X
```

```
120 NEXT Y
130 GOTO 130
```

Zodra het programma is gestart, wordt het beeldscherm zwart. Druk op één van de cursortoetsen en de beeldstip trekt een lijn in de overeenkomstige kleur. Regel 90 zorgt ervoor dat bij het indrukken van een verkeerde toets (de toets met de pijl omhoog of een combinatie van toetsen) de laatst geselecteerde kleur gehandhaafd blijft. Het lijkt onwaarschijnlijk, maar het zojuist ingetypte programma is op software-gebied vrijwel het enige dat nodig is om afbeeldingen gedigitaliseerd in het videogeheugen op te slaan.

De analoog-naar-digitaal-omzetter

Eigenlijk verdient de schakeling die we hier gaan beschrijven de naam ADC niet helemaal.
Omdat we slechts drie lichtniveaus willen detecteren, hebben we genoeg aan twee vergelijkingsschakelingen (comparators). Afbeelding 21-2 laat het schema van de comparator zien.



Afbeelding 21-2 De comparator.

Aan één van de ingangen van de beide comparators wordt het door de leeskop afgegeven signaal aangeboden, terwijl op de andere ingang een referentiespanning staat. Door de referentiespanningen voor beide comparators apart instelbaar te maken, kunnen we twee niveaus vastleggen waarbij de uitgangsspanning omklapt. Ligt het signaal onder de twee niveaus, dan zijn beide com-paratoruitgangen laag. Tussen de twee niveaus in is één ervan hoog terwijl de andere laag is. Bij een ingangssignaal dat zich boven het tweede niveau bevindt, zijn beide uitgangen hoog.
Twee relais, die direct door de schakeling worden gestuurd, nemen de functie van de joystick-schakelaars over.

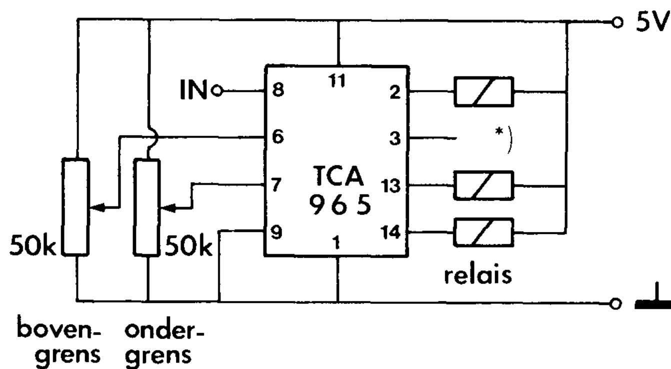
Deze schakeling heeft een aardige eigenschap: we kunnen volstaan met het gebruik van slechts twee aansluitingen van de joystick-ingang. Bijvoorbeeld pen 1 en pen 4, die overeenkomen met omhoog en rechts. Het schema uit afbeelding 21-2 laat zien dat, wanneer geen relais aanspreekt, STÏCK(1) de waarde 0 oplevert. Als het onderste relais aanspreekt, heeft STICK(1) de waarde 1 en als beide relais aanspreken (pen 1 en pen 4 aan massa) krijgt STICK(1) de waarde 2 (schuin rechts omhoog). In een programma vertalen we deze waarden naar kleuren.

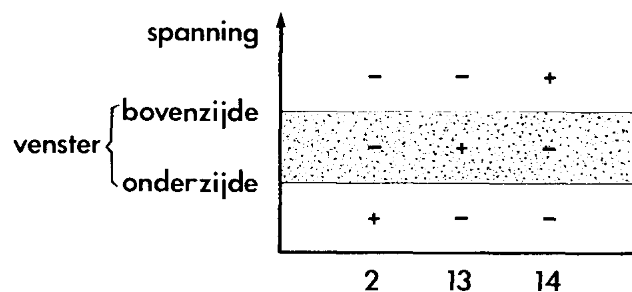
```
IF STICK(1)=0 THEN COLOR 6 (donkerrood)
IF STICK(1)=1 THEN COLOR8 (rood)
IF STICK(1)=2 THEN COLOR 9 (lichtrood)
```

In groen ziet dat er vergelijkbaar uit:

```
IF STICK(1)=0 THEN COLOR 12  (donkergroen)
IF STICK( 1 )=1 THEN COLOR   2  (groen)
IF STICK(1)=2 THEN COLOR   3  (lichtgroen)
```

De besproken schakeling is eigenlijk een vensterindicator. Een vensterindicator bekijkt of een spanning onder, in, of boven een vooraf ingesteld venster ligt. Deze functie komt in de meet-en-regeltechniek veel voor als bewaker van processen. In zo'n proces kan het voorkomen dat een temperatuur zich tussen twee uitersten moet bevinden. Of dat van een voorwerp de afmetingen tussen zekere grenzen moeten liggen. Een vensterindicator kan dan 'goedgekeurd' (binnen de grenzen) of 'afgekeurd' (buiten de grenzen) aangeven. Vanwege het veelvuldige gebruik van vensterindicatoren is het niet verwonderlijk dat er een IC bestaat dat de functie van vensterindicator zelfstandig kan vervullen. De TCA 965 (Siemens) is zo'n IC. Afbeelding 21-3 laat één van de manieren zien waarop dit IC te gebruiken is.
Een bijkomend voordeel van dit IC is de ingebouwde hysteresis. Kleine variaties van het ingangssignaal in de buurt van een schakelpunt resulteren hierdoor niet in een storend geratel van de aangesloten relais. We passen relais toe, ondanks het feit dat een directe koppeling zoveel eenvoudiger is. De reden hiervoor kunt u raden. Een directe koppeling met het binnenste van de computer is ongewenst omdat het, bij een eventuele storing, niet erg plezierig is als een gedeelte van de computer wordt opgeblazen.
Om het hier beschreven project ook voor de computer- hardware-leek geschikt te laten zijn, zullen we elk risico bij dit project dan ook zorgvuldig mijden en gebruik maken van reed-relais. Reed-relais zijn klein, passen in een IC-voet en kunnen rechtstreeks vanuit het venster-IC worden gestuurd. Ze voldoen bovendien aan de door ons gespecificeerde maximale schakelfrequentie van



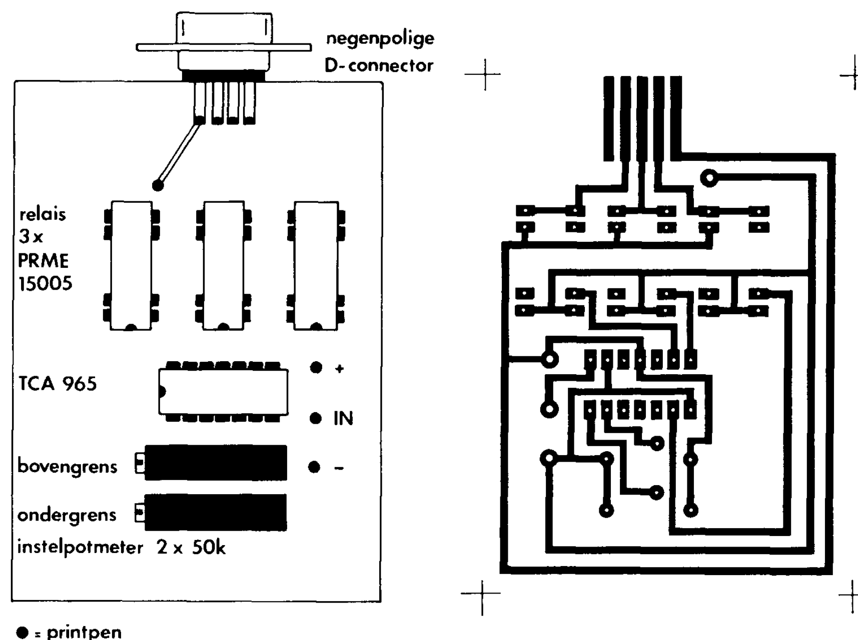


Afbeelding 21-3 Het gebruik van de vensterindicator TCA 965.
**) Uitgang 3 van de TCA 965 geeft aan als het ingangssignaal zich buiten het venster bevindt en wordt door ons niet gebruikt.*

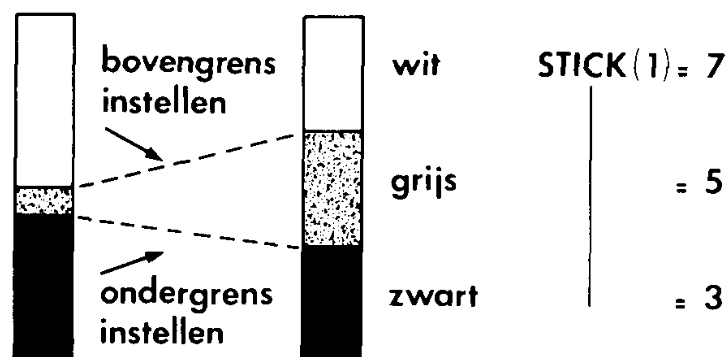
honderd hertz. De print-layout laat zien dat de joystick-plug op de print wordt bevestigd. Voor dit doel is een chassisdeel uitstekend geschikt. Chassisdelen in de gewenste uitvoering zijn algemeen verkrijgbaar in de serie D-connecto-ren. Bij sommige MSX-computers ligt de joystick-ingang te diep in de kast om een chassisdeel te kunnen gebruiken. Voor die computers is een stekker aan een kabeltje de oplossing.
Omdat we het gebruik van dubbelzijdige printen willen vermijden, gebruiken we de pennen 2, 3 en 4 van de joystick-aansluitplug. Alleen voor de massaaansluiting is dan een draadbrugje nodig.
Zoals de print aangeeft, betrekken we de 5 V-voedingsspanning rechtstreeks uit de computer. In totaal trekt de schakeling iets meer dan 20 mA. Dit levert voor de voeding van de MSX-computer geen enkel probleem op. Omdat we op deze wijze toch met (in dit geval minder kwetsbare) computer-elektronica spelen, is het zaak om ook deze schakeling eerst met een externe 5 V-voeding uit te proberen. De eerder genoemde batterij met een 5 V-regelaar kan ook nu weer voortreffelijke diensten bewijzen.

De afregeling

Als de schakeling is aangesloten op een 5 V-voeding, is de afregeling ervan kinderlijk eenvoudig. We sluiten de sensor aan en zetten de potentiometers in de middenstand. Als we nu met een wit papiertje de leeskop tot op ongeveer



twee millimeter naderen, schakelen de reed-relais: we horen ze tikken. Bij een bepaalde afstand tussen de sensor en het papier vibreren de reed-relais: het venster-IC detecteert een grenswaarde en twijfelt tussen het wel of niet aanspreken van het relais.
We zoeken wat foto's in een krant op. Misschien ligt de krant er nog van de vorige test, toen de werking van de sensor werd beoordeeld. We zetten de sensor op een wit gedeelte van de krant en meten de uitgangsspanning. Vervolgens plaatsen we de sensor op een gemiddeld grijs gedeelte en meten ook nu weer de uitgangsspanning. Tot slot meten we de uitgangsspanning als de sensor op een zwart vlak staat.
Stel de potentiometers dusdanig in, dat de reed-relais aanspreken zoals in afbeelding 21-5 grafisch staat weergegeven.

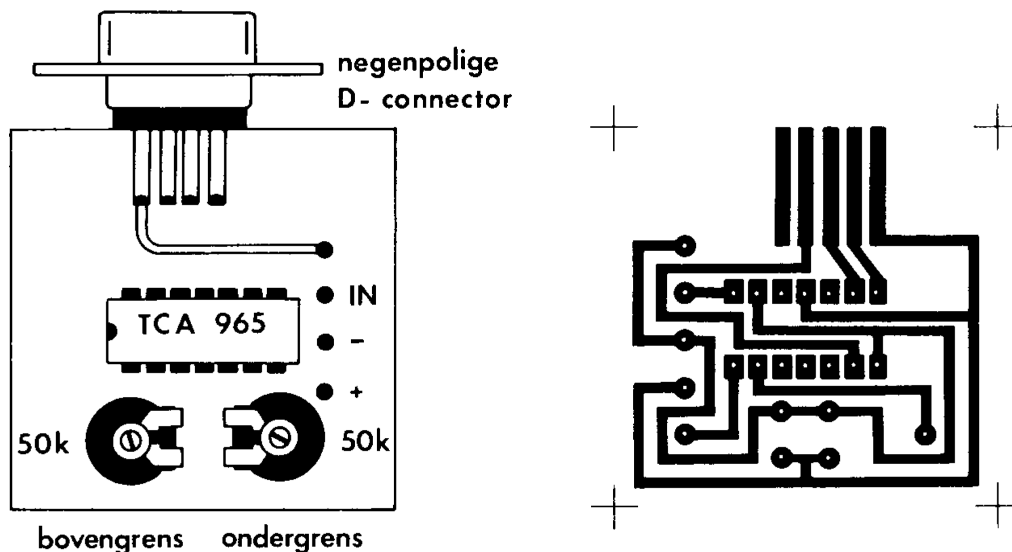


Afbeelding 21-5 De instelling van de potentiometers.

Deze afregeling voldoet voorlopig. Eventueel kunnen we in een later stadium de potentiometers nog een beetje bijstellen, afhankelijk van de aard van de op te nemen afbeelding.

22 Een schakeling voor gevorderde elektronici

Omdat u dit hoofdstuk leest, nemen we aan dat u een vaste hand van solderen heeft en bovendien de nodige kennisvan elektronica. Op basis hiervan kunnen we de ADC nog aanzienlijk vereenvoudigen. De reed-relais, die in het vorige hoofdstuk een plaatsje op de ADC-print hebben gekregen, zitten daar uitsluitend om risico's voor de beginnende elektronicus te vermijden. De uitgangen van het besproken venster-IC liggen in de actieve toestand aan massa. Ze verrichten dus dezelfde functie als een joystick-schakelaar. Het printje komt er aantrekkelijk eenvoudig uit te zien als de relais vervallen. Vanwege de betrouwbaarheid van het IC, de zeer geringe belasting ervan voor de computer en het feit dat de joystick-ingang in het ergste geval - als werkelijk alles verkeerd gaat- alleen maar aan de voedingsspanning van de computerzelf komt te liggen, lopen we geen echt risico als de vensterindicator rechtstreeks met de joystick-ingang wordt verbonden.



Afbeelding 22-1 De vereenvoudigde comparatorschakeling: print-layout en componentenopstelling.

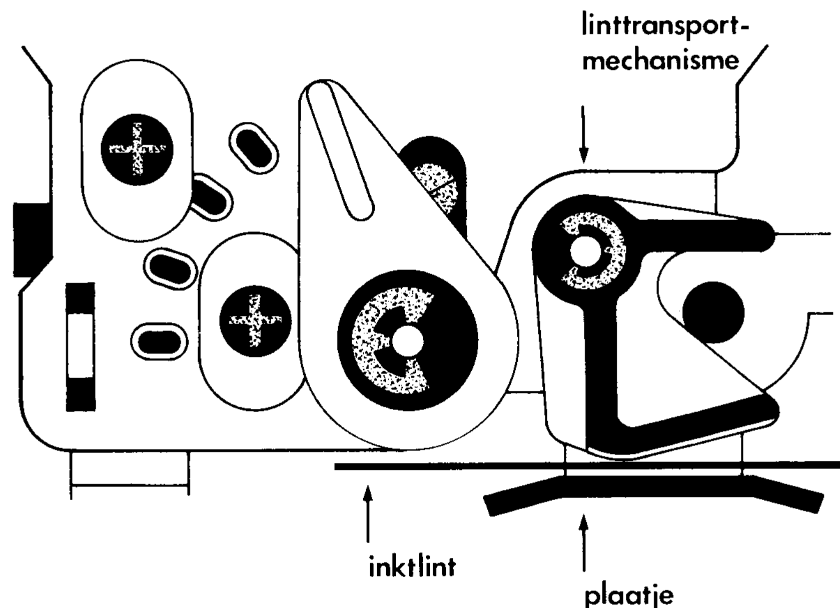
Het is zelfs de vraag of voor deze eenvoudige schakeling wel een speciaal geprepareerd printje nodig is. In een kwartiertje zetten we de hele schakeling op een stukje gaatjesprint in elkaar.
Als de ruimte in de printer die voor de experimenten gebruikt gaat worden dat toelaat, is de gedachte om de opnemer en de vensterindicator op een printje te combineren erg aantrekkelijk. Een soepel snoertje met stekker, zoals dat aan een joystick zit, geeft het geheel een professioneel uiterlijk. Het uiterlijk kan nog verder verfraaid worden door het geheel in een geschikte kunststoffen behuizing te bouwen.

23 Printeraanpassing

Het is niet de bedoeling dat de printer ingrijpend wordt omgebouwd. Als u in de luxe omstandigheid verkeert in het bezit te zijn vaneen printer die zijn beste tijd heeft gehad, kan de aanpassing natuurlijk definitief worden doorgevoerd. We zullen niet gaan knutselen aan de printer zelf en verrichten zelfs geen handelingen die een eventueel nog lopende garantieperiode in gevaar kunnen brengen.
In normale gevallen is de printer op de computer aangesloten via de printerkabel. De computer stuurt gecodeerde tekens over de kabel naar de printer. Zodra de printerbuffer een volle regel heeft opgenomen, stuurt de printer het 'busy'-signaal naar de computer, om aan te geven dat de computer moet wachten met het verzenden van nieuwe tekens. De printkop gaat lopen en de regel wordt afgedrukt. In de tijd dat de printer de regel afdrukt, staat de computer op non-actief. Bij sommige printers kan de buffer meer dan één regel opslaan en kan de computer tijdens het printen gewoon doorwerken. We beschouwen hier een eenvoudige matrixprinter met een buffer van slechts één regel. Tijdens de beweging van de printkop willen we de lichtintensiteit langs die regel meten en opslaan. Dit is echter nogal moeilijk als de printer tijdens deze slag de computer uitschakelt. We kunnen het 'busy' uitschakelen door de desbetreffende draad in één van de aansluitpluggen van de printerkabel los te nemen. In dat geval moeten we toch nog iets doen om de juiste spanning op de losgekoppelde aansluitpunten te krijgen. Het zonder meer doorknippen van een kabel-ader geeft niet het gewenste resultaat. We blijven bij ons uitgangspunt en laten ook de kabel intact. Sterker nog, we sluiten de printerkabel helemaal niet aan!
Vrijwel iedere printer kan een zelftest uitvoeren. Tijdens een zelftest worden alle tekens die de printer kent, afgedrukt. Na elke regel wordt bovendien het papier een regel verder gedraaid; een proces dat zich steeds herhaalt. De door ons gewenste scanbeweging wordt tijdens de zelftest dus netjes uitgevoerd, zonder dat de computer geblokkeerd raakt. We kunnen de over te nemen afbeelding op het kettingpapier plakken en het zelftestmechanisme in werking stellen. Op een kleinigheid na... het is niet de bedoeling dat de met zorg gekozen afbeelding vol met tekens wordt bedrukt. We kunnen dit op twee manieren voorkomen. We kunnen de elektrische verbinding naarde printkop losnemen; bij vrijwel alle printers is dat een tamelijk eenvoudige zaak. Mocht dat echter bij sommige printers minder eenvoudig blijken te zijn, dan verwijderen we eenvoudigweg het inktlint, zodat de printer geen zichtbare sporen op de afbeelding achterlaat. Wordt dit laatste door de printerfabrikant afgeraden, dan zit er niets anders op dan tussen het lint en het papier waarop de te kopiëren afbeelding zich bevindt een papiertje te schuiven, zodat de printtekens hierop afgedrukt worden in plaats van op de over te nemen afbeelding.

De mechanische aanpassing

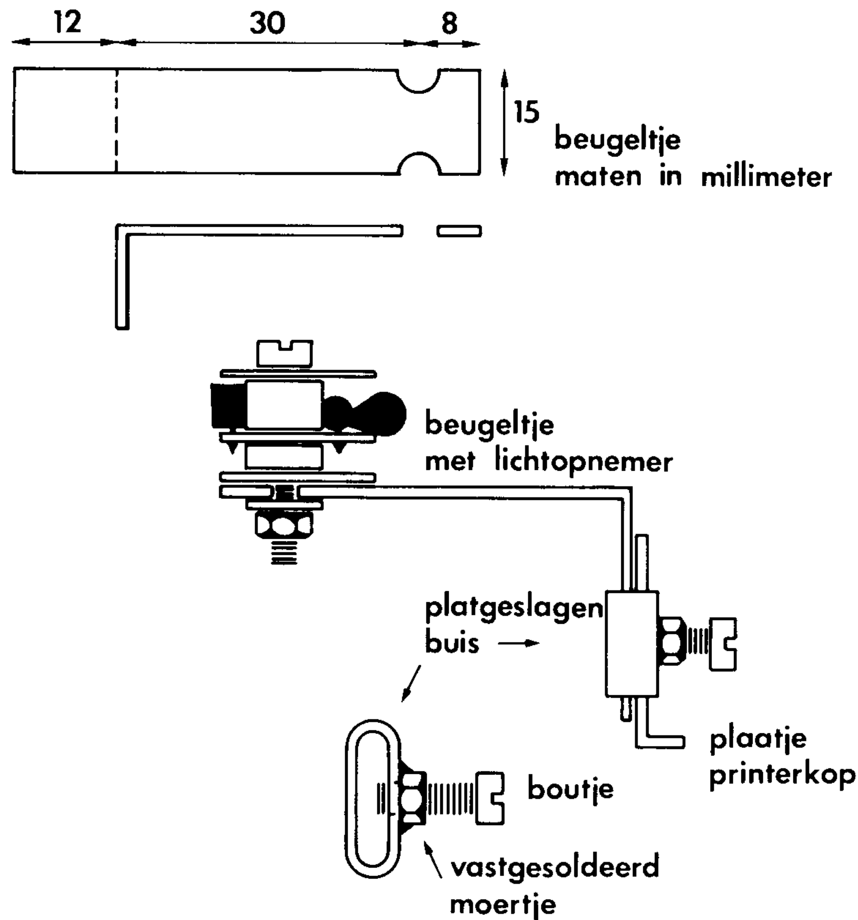
Het is te veel gevraagd om in dit boek voor alle printers te beschrijven hoe de leeskop het best op het bewegende deel van de printer kan worden bevestigd. Met enige inventiviteit is er in de meeste gevallen wel een oplossing te vinden. Voor een populaire printer geven we hier een oplossing. Voor andere printers kan een kleine aanpassing tot het gewenste effect leiden. In de echt moeilijke gevallen kan de leeskop ook met dik dubbelzijdig plakband worden bevestigd. Ook een stukje plasticine (boetseerklei) kan voor de oplossing in nood zorgen. De GP-100 van Seikosha is een robuuste matrixprinter die bij veel computerbezitters wordt aangetroffen. Afbeelding 23-1 laat zien dat het inktlint op de terugweg langs een haaks omgezet metalen plaatje wordt geleid. Ertegenover zit een palletje dat ervoor zorgt dat het lint tijdens het printen naar één kant wordt meegenomen.



Afbeelding 23-1 De printerkop van de GP-100 van Seikosha.

Het plaatje vormt een stevig geheel met de rest van de constructie en is er als het ware voor gemaakt om de leeskop aan te bevestigen. Voor dit doel hebben we twee hulpstukjes nodig. In de eerste plaats een metalen beugeltje, dat bijvoorbeeld uit een strookje aluminium wordt gemaakt. Eventueel kan een kant en klaar beugeltje uit de ijzerwinkel of hobbyzaak op maat worden gezaagd. De breedte ervan is niet erg belangrijk, mits die niet méér dan anderhalve centimeter bedraagt.
Afbeelding 23-2 laat zien hoe de leeskop op het beugeltje wordt bevestigd met behulp van de M3-boutjes op de leeskop. Het tweede hulpstukje is een metalen cilindertje met een hoogte van één centimeter (iets minder mag ook) en een diameter van ongeveer 15 millimeter. Zeer geschikt hiervoor is een stukje

koperen waterleidingbuis. We drukken het stukje buis plat en boren er vervolgens een gaatje in. Met een M3-tapje kunnen we schroefdraad in het gat aanbrengen. We moeten in dit geval een gaatje boren met een diameter van 2,5 millimeter. Mocht u geen tapje hebben, dan kunt u ook een M3-moertje op de platgeslagen buis solderen. Het gat waarvoor het moertje komt te zitten, moet dan een diameter hebben van 3 of 3,5 millimeter.



Afbeelding 23-2 De bevestiging van de lichtopnemer met het beugeltje en het hulpstuk om het beugeltje met de lichtopnemer op de printerkop te bevestigen.

Let er goed op dat de leeskop vrijloopt van papiergeleiders en andere obstakels. Doordat de printslede iets verend is opgehangen, rust de leeskop met een lichte druk tegen het papier. De afstand tot het papier is hierdoor zeer stabiel. Sommige printers hebben geen zelftestmogelijkheid die van buitenaf te starten is. Bij de GP-100 kunnen we deze mogelijkheid zelf installeren door de punten 15 en 35 van de aansluitplug (aan de achterzijde van de printer) met elkaar te verbinden. Een miniatuurschakelaartje dat in een plug wordt gemonteerd, verricht deze klus op eenvoudige wijze. De rasechte werktuigbouwkundige kan het schakelaartje ook in de printer monteren.

We zijn er bijna, en zijn nu zover dat we kunnen proefdraaien. Hiervoor brengen we in het al besproken programma nog enkele wijzigingen aan.

```
10 REM ongesynchroniseerd
20 SCREEN 3:COLOR 1,1,1:CLS
30 FOR Y=0 TO 192 STEP 4
40 FOR X=0 TO 255 STEP 4
50 K=STICK(1)
60 IF K=0 THEN C = 9
70 IF K=5 THEN C=8
80 IF K=6 THEN C=6
90 PSET(X,Y),C
100 NEXT X
110 A$=INPUT$(1)
120 NEXT Y
130 GOTO 130
```

Breng een afbeelding in de printer door deze op het kettingpapier te plakken. Plak de bovenkant met een strookje plakband vast. Neem hiervoor niet-reflecterend plakband, zoals bijvoorbeeld Scotch-tape. Zet ook de zijkanten van de afbeelding met een stukje plakband vast. Let erop dat de leeskop geen loszittende randjes papier mee kan nemen. Gebruik daarom indien mogelijk een kopie op A4-formaat. Kies bij voorkeur een afbeelding met hoge contrasten. Zet de printer op zelftest en start het programma (RUN [Return] of F5). Telkens als de printer is teruggekeerd naar de uitgangspositie, moet u de spatiebalk even indrukken. Op het scherm verschijnt het eerste beeld. Misschien nog wat onherkenbaar omdat de synchronisatie met de hand niet erg optimaal is. Daar gaan we nu wat aan doen.

De synchronisatie

Het synchronisatieprobleem lossen we op met een microschakelaar die is ingedrukt als de leeskop zich in de uitgangspositie (links) bevindt. Een geschikt plaatsje voor die schakelaar is altijd wel te vinden. Bij de GP-100 is de steun waarover de linker inktlintcassette wordt geschoven, uitstekend geschikt. Zelfs het gat zit op de goede plaats. Met een afstandsbusje wordt de microschakelaar zó gemonteerd dat hij aan het begin van de slag door de leeskop wordt ingedrukt. Bij sommige printers kan de schakelaar met het al eerder genoemde dubbelzijdige plakband op een geschikte plaats worden aangebracht. Neem een microschakelaar met een lichte schakeldruk en let erop dat het schakelarmpje lang genoeg is. We gebruiken het maakcontact. Eén van de schakelcontacten wordt door een lange draad met de massa van de leeskop verbonden. De draad moet de volledige printslag kunnen volgen. Het andere contact van de schakelaar verbinden we met pen 6 van de joystick-plug. Deze pen is de triggeringang en bevindt zich aan de componentenzijde van de vensterindicatorprint. Met STRIG(0) kan worden bepaald of de schakelaar al dan niet is ingedrukt.

In het programma nemen we de automatische synchronisatie op in een paar aanvullende regels.

```
10 REM gesynchroniseerd
20 SCREEN3:COLOR 1,1,1:CLS
30 IF STRIG(l) THEN 30 ELSE 40
40 FOR Y=0 TO 192 STEP 4
50 FOR X=0 TO 255 STEP 4
60 K=STICK(1)
70 IF K=3 THEN C=9
80 IF K=5 THEN C=8
90 IF K=7 THEN C=6
100 PSET(X,Y),C
110 NEXT X
120 ÏF STRIG(l) THEN 130 ELSE 120
130 IF STRIG(l) THEN 130 ELSE 140
140 NEXT Y
150 GOTO 150
```

Als de 256 punten langs de horizontale as in stappen van vier pixels (regel 50) zijn afgetast (regel 110), zorgt regel 120 ervoor dat het programma pas doorgaat als de schakelaar is ingedrukt. Regel 130 zorgt ervoor dat de volgende lijn op het scherm verschijnt als de leeskop vrijkomt van de schakelaar. Het begin van elke beeldlijn op het scherm komt door de synchronisatie steeds weer nauwkeurig overeen met de kantlijn van het papier. In het volgende hoofdstuk zullen we nog enkele verfijningen in het programma aanbrengen.

24 Het werken met de digitizer

In de voorafgaande hoofdstukken zijn de hardware-zaken aan de orde geweest. In dit hoofdstuk gaan we verder in op de software-kant van het gebeuren.
Het zal u waarschijnlijk al zijn opgevallen dat de synchronisatie af en toe een steekje laat vallen. Toch lijkt het programma waterdicht. De computer maakt geen fouten. In dit geval doet hij zijn werk zelfs te goed. Door de hoge frequentie waarmee de eindschakelaar wordt uitgelezen (met de STRIG(1) opdracht), kan het gebeuren dat er al weereen meting wordt gedaan als de contacten van de schakelaar nog nadenderen. De computer interpreteert dat alsof de leeskop de uitgangspositie alweer heeft verlaten, terwijl die nog maar net gearriveerd is. Het beeldpunt begint dan ook alvast over het scherm te lopen, terwijl de leeskop nog rustig in de hoek staat. Voor dit probleem zijn meerdere oplossingen te verzinnen: we kunnen een condensator over de schakelaar plaatsen of we kunnen een vertraging in het programma aanbrengen:

```
125 FOR Q=l TO 100:NEXT Q
```

Het eenvoudigst is niet het maakcontact, maar het verbreekcontact van de microschakelaarte gebruiken en de regels 30,120 en 130 hierop aan te passen.

```
30  IF STRIG(l) THEN 40 ELSE 30
120 IF STRIG(l) THEN 120 ELSE 130
130 IF STRIG(l) THEN 140 ELSE 130
```

Het contact-dender-probleem is nu verlegd naar de start van de leescyclus. De tijdverschillen die hier optreden, zijn echter niet waarneembaar. Echter, bij minder goede schakelaars verloopt ook het verbreken van het contact niet helemaal betrouwbaar. Om het zekere voor het onzekere te nemen, bouwen we toch maareen wachtlus in en maken gewoon gebruik van het maakcontact.

```
10 REM gesynchroniseerd met wachtlus
20 SCREEN 3:COLOR 1,1,1:CLS
30 IF NOT STRIG(l) THEN 40 ELSE 30
40 FOR Y=0 TO 192 STEP 4
50 FOR X=0 TO 255 STEP 4
60 K=STICK(1)
70 IF K=3 THEN C=9
80 IF K=5 THEN 0=8
90 IF K-7 THEN C=6
100 PSET(X,Y),C
110 NEXT X
120 IF NOT STRIG(l) THEN 120
130 FOR Q=l TO 100:NEXT Q
140 IF STRIG(l) THEN 140
150 NEXT Y
160 GOTO 160
```

Dan lijkt nu alles rond. Regel 160 zorgt ervoor dat het beeld netjes op het scherm blijft staan totdat we CONTROL-STOP indrukken of de computer uitzetten. Dus tot zolang genieten maar van het mooie plaatje... Maar daarvoor hebben we alle moeite niet gedaan. We willen op zijn minst het plaatje bewaren om het ineen programma als versiering te gebruiken. De spelletjesfanaten zullen de opgenomen beelden willen gebruiken als decor voor de meest spannende avonturen. De combinatie van sprites en opgenomen achtergrondbeel-den is natuurlijk om van te watertanden. Door sprites op de juiste plaatsen te laten verdwijnen achter een sprite die de achtergrondkleur heeft, kunnen we zelfs de suggestie opwekken dat het beeld ruimtelijk is. De zaak is duidelijk: het beeld moet bewaard blijven, in welke vorm dan ook.
Voor de bezitters van een 3,5 schijfeenheid is er een gemakkelijke oplossing voorhanden. Een kleine programma- uitbreiding zorgt ervoor dat het volledige videogeheugen op schijf wordt gezet.

```
10  REM gesaved op schijf
20  SCREEN 3:COLOR 1,1,1:CLS
30  IF STRIG(l) THEN 40 ELSE 30
40  FOR Y=0 TO 192 STEP 4
50    FOR X=0 TO 255 STEP 4
60      K=STICK(1)
70      IF K=3 THEN C=9
80      IF K=5 THEN C=8
90      IF K=7 THEN C=6
100    PSET(X,Y),C
110   NEXT X
120   IF STRIG(l) THEN 120 ELSE 130
130   FOR Q=l TO 100:NEXT Q
140   IF STRIG(l) THEN 150 ELSE 140
150 NEXT Y
160 BSAVE "BEELD1",0,16383,S
```

Een programma om de beelden terug op het scherm te zetten, is al even eenvoudig.

```
10 REM recover
20 BLOAD "BEELD1",S
30 GOTO 30
```

Let erop dat de naam die we voor het beeld kiezen (in dit geval 'BEELD') slechts éénmaal op een diskettekant mag voorkomen. Voor elk beeld dat moet worden opgeslagen, moet u een andere naam verzinnen. De beschreven opslagmethode is eenvoudig en geeft de volledige vrijheid het

beeld vanuit een programma op te roepen. Het laden van het beeld vraagt enige seconden, maar de beeldvorming zelf geschiedt zeer snel. Er bestaat ook een manier om de beelden op een cassetteband op te slaan waarbij eventuele beeldbewerking achteraf zeer gemakkelijk is. De mogelijkheden hiertoe worden in volgende hoofdstukken onder de loep genomen. Eerst poetsen we het beeld nog wat op. Het zou erg toevallig zijn als de lijnsnel-heid op het scherm precies overeen zou komen met de snelheid waarmee de leeskop over het papier beweegt. In de meeste gevallen blijft de leeskop wat achter. Door de hogere schrijfsnelheid op het beeldscherm wordt het beeld in horizontale richting uitgerekt. We laten de computer in de pas lopen door het programma wat af te remmen. We doen dat door het inbouwen van een welbekende FOR-NEXT-wachtlus, direct nadat een punt is ingekleurd. Als de printer aan het einde van de regel is gekomen, zal de computer - vanwege de ingebouwde vertraging - nog niet aan het einde van het scherm zijn. De gegevens die de leeskop vanaf dat ogenblik doorgeeft, zijn echter niet meer relevant: de informatie die de leeskop tijdens de terugslag opneemt, hoort niet bij het op te nemen beeld. Behalve het vertragen van de beeldvorming maken we daarom ook de horizontale lijnen wat korter, door de X-coördinaat niet van O tot 255 te laten lopen, maar van 50 tot 200.

```
10  REM snelheidscorrectie
20  SCREEN 3:COLOR 1,1,1:CLS
30  IF STRIG(1) THEN 40 ELSE 30
40  FOR Y=0 TO 192 STEP 4
50   FOR X=50 TO 200 STEP 4
60     K=STICK(1)
70     IF K=3 THEN C=12
80     IF K=5 THEN C=2
90     IF K=7 THEN C=3
100    PSET(X,Y),C
110     FOR Q=1 TO 12:NEXT Q
120   NEXT X
130   IF STRIG(1) THEN 130 ELSE 140
140   FOR Q=1 TO 100:NEXT Q
150   IF STRIG(1) THEN 160 ELSE 150
160 NEXT Y
170 BSAVE  "BEELD1",0,16383,S
```

Hoe bevallen de nieuwe kleurtjes in de regels 70 tot en met 90? De wachtlus in regel 110 en de coördinatenaanpassing in regel 50 kunnen per printer wat verschillen. U kunt ze naar wens aanpassen. Het beeld kan er ook opzettelijk mee worden vervormd. Op het onderwerp beeldbewerking komen we in een volgend hoofdstuk uitgebreid terug.

25 Beeldopslag

In het vorige hoofdstuk hebben we gezien hoe het gedigitaliseerde beeld vanuit het videogeheugen in één keer op een 3,5 inch-schijf kan worden gezet. Nu heeft niet iedereen een 3,5 inch-schijfeenheid. De cassetterecorder is nog steeds een veel gebruikt opslagmedium, maar ook de 'echte' diskdrive ligt inmiddels binnen het budget van menig computergebruiker. We zoeken een methode waarbij de beeldpunten goed toegankelijk zijn voor verdere bewerking. In dit hoofdstuk gaan we daarom de beeldopslag drastisch wijzigen.
Van elk beeldpunt leggen we de X-coördinaat, de Y-coördinaat en de kleur vast. Door het slim aan te pakken kan dat op een zeer elegante manier. Het scherm waarop we werken (screen 3) heeft op de X-as (horizontaal) 64 posities. Op de Y-as (verticaal) treffen we 48 lijnen aan. Elke combinatie van X en Y wijst een beeldpunt aan. Elk beeldpunt heeft een eigen kleur. In de voorafgaande programma's werd de kleur aangeduid met een C. We blijven dit consequent doen, maar brengen een kleine uitbreiding aan. Behalve de kleuraan-duiding worden ook de coördinaten, dus de plaats op het beeldscherm opgenomen.

```
C(X,Y)
```

De kleuren van de beeldpunten op het hele beeldscherm liggen vast met deze variabele. Zo zijn de kleuren van alle punten op regel 7 vastgelegd met C(X,7). Het derde beeldpunt op die regel heeft de kleur C(3,7). Als alle beeldpunten op één regel moeten worden afgebeeld, kan dat door X van 0 tot 63 (in totaal 64 beeldpunten) te laten lopen.

```
10 REM regel
20 FOR X=0 TO 63
30 PSET(X,7),C(X,7)
40 NEXT X
```

Om alle punten op het beeldscherm in te kleuren, dient bovendien de Y-coördinaat het traject van O tot 47 (in totaal 48 regels) te doorlopen.

```
10 REM scherm
20 FOR Y=0 TO 47
30   FOR X=0 TO 63
40    PSET(X,Y),C(X,Y)
50   NEXT X
60 NEXT Y
```

In het inleesprogramma dat in het vorige hoofdstuk tot stand is gekomen, gaan we de variabele C vervangen door C(X,Y). We mogen dat echter niet zonder meer doen. Arrays (in dit geval alle variabelen C(X,Y) bij elkaar) moeten van tevoren worden aangekondigd. De computer wil graag weten hoeveel geheugenruimte er voor arrays moet worden gereserveerd, omdat ze een zeer grote omvang kunnen aannemen. In ons geval gaat het om een array met 64 x 48=3072 elementen. De DIM-opdracht reserveert geheugenruimte voor de array.

```
DIM C(63,47)
```

N.B.: De kleur van het punt in de linker bovenhoek wordt door C(0,0) bepaald. De oplettende lezer zal opmerken dat het beeldopnameprogramma niet de volle breedte van het scherm benut. We zouden dus kunnen volstaan met het niet tot het einde van de regel door laten lopen van de X-coördinaat. Een MSX-computer met 64K vrij geheugen heeft echter voldoende ruimte om alle beeldpunten op te bergen. En, in een later stadium gaan we het hele scherm bij beeldbewerking betrekken.
Het langzaam groeiende programma wordt geschikt gemaakt voor de toepassing van een array waarin de kleuren van de beeldpunten komen te staan:

```
10  REM beeldpunten inlezen
20  SCREEN 3:COLOR 1,1,1:CLS:KEY OFF
22  DIM C(63,47)
30  IF STRIG(1) THEN 40 ELSE 30
40  FOR Y=0 TO 47
50   FOR X=12 TO 50
60     K-STICK(1)
70      IF K=3 THEN C(X,Y)=12
80     IF K=5 THEN C(X,Y)=2
90     IF K=7 THEN C(X,Y)=3
100    PSET(4*X,4*Y),C(X,Y)
110     FOR Q=1 TO 6:NEXT Q
120   NEXT X
130   IF STRIG(1) THEN 130 ELSE 140
140   FOR Q=1 TO 100:NEXT Q
150   IF STRIG(1) THEN 160 ELSE 150
160 NEXT Y
```

Let vooral op de regels 100 en 110. De STEP4-opdracht is vervallen. De factor vier is nu in regel 100 terug te vinden. Omdat regel 100 enige rekentijd vraagt, is de wachtlus in regel 110 iets ingekort om het beeld en de printer weer op elkaar af te stemmen. Bovendien is de laatste regel, waarin het beeld op schijf werd vastgelegd, vervallen. In plaats daarvan bouwen we het beeld direct weer op door de toevoeging van een nieuw stukje programma:

```
170 CLS
180 FOR Y=0 TO 47
190  FOR X=0 TO 63
200    PSET(4*X,4*Y),C(X,Y)
210  NEXT X
220 NEXT Y
230 GOTO 230
```

Vergelijk regel 50 eens met regel 190. In regel 50 lezen we het beeld in. Om het beeld netjes in het midden van het scherm te plaatsen en geen punten in te kleuren met 'terugslaginformatie', loopt de telling langs de horizontale as van 12 tot 50. In regel 190 daarentegen plaatsen we het beeld weer op het scherm. Alle punten lager dan X=12 en hoger dan X=50 zijn tijdens het opnemen niet gekleurd en hebben daarom de kleur 0: transparant. Dat wil zeggen dat ze op het scherm verschijnen als zwart, de achtergrondkleur. De beeldpunten worden één voor één weer in de juiste kleur op het scherm gezet. Voor het opslaan van de afzonderlijke beeldpunten kunnen we het zojuist toegevoegde programmadeel wijzigen.

```
170 OPEN "CAS:COPIE" FOR OUTPUT AS #1
180 FOR Y=0 TO 47
190 FOR X=0 TO 63
200 PRINT #1,C(X,Y)
210 NEXT X
220 NEXT Y
230 CLOSE
```

Let er weer op dat de X-coördinaat de gehele lijn van 0 tot en met 63 doorloopt. Regel 170 opent het bestand op de cassetteband dat bestemd is om er gegevens in op te bergen (FOR OUTPUT). Het is natuurlijk mogelijk om elk ander opslagmedium te gebruiken. Voor een 3,5 inch-schijf zou regel 170 er als volgt uitzien:

```
170 OPEN "COPIE" FOR OUTPUT AS #1
```

Als de gegevens zijn weggeschreven, kan vrijwel hetzelfde programmadeel worden gebruikt om ze weer tevoorschijn te halen.

```
5  REM reproduktie
10 SCREEN3:COLOR 1,1,1:CLS
20 OPEN"CAS:COPIE" FOR INPUT AS #1
30 FOR Y=0 TO 47
40   FOR X=0 TO 63
50     INPUT #1,C(X,Y)
60    PSET(4*X,4*Y),C(X,Y)
70   NEXT X
80 NEXT Y
90 CLOSE
```

In regel 20 van het reproduktieprogramma wordt het bestand weer geopend, nu met de bedoeling er gegevens uit te halen (FOR INPUT). Het is erg onhandig als alle opgeslagen beelden met dezelfde naam (COPIE) op een cassettebandje staan. Op een 3,5 inch-schijf is dat zelfs niet eens toegestaan. Er bestaat een goede methode om een bestand te openen met een naam naar keuze. We doen dat door de naam apart op te vragen en de bestandsnaam samen te stellen uit een vast deel en een gedeelte dat steeds weer anders kan zijn.

```
10 INPUT "NAAM";N$
20 OPEN "CAS:"+N$+".DAT"
```

Hetzelfde geldt voor de 3,5 inch-schijf:

```
10 INPUT "NAAM";N$ 20 OPEN N$+".DAT"
```

Probeer één van deze twee miniprogramma's maar eens en vul voor de naam COPIE 1 in. Als alles goed gaat, komt er op het opslagmedium een leeg bestand te staan met de naam COPIE1.DAT.
Zoals bij de MSX-gebruiker bekend behoort te zijn, mag de naam uit niet meer dan acht tekens bestaan, terwijl het eerste teken een letter moet zijn. De drie-letterige extensie kan weliswaar vrij worden gekozen, maar voor de herkenbaarheid is het goed om een bepaalde norm aan te houden. Zo wordt de toevoeging .DAT in het algemeen gebruikt voor DAT A-bestanden. Als u de herkenbaarheid van de bestanden wilt verbeteren door .COP .DUP of iets dergelijks te gebruiken, is dat uiteraard geoorloofd, mits het voor uzelf maar herkenbare toevoegingen zijn. In dit boek zullen de toevoegingen .PIX en .SCR worden gebruikt voor beelden die als kleur-arrays (C(X,Y)> respectievelijk video-geheugen-dumps zijn opgeslagen.
Nu het opslaan van gegevens op tape of diskette geen problemen meer hoeft op te leveren, wordt het zo langzamerhand tijd om het beeldopnameprogram-ma in een goed hanteerbare vorm te gieten. Een aanzienlijk deel van het programma is al ingetypt, zodat volstaan kan worden met de toevoegingen. Met RENUM wordt de regelnummering weer op orde gebracht.

```
20 REM kopieer programma
30 KEY OFF:COLOR 1,10,10:CLS
40 PRINT"Breng een afbeelding in de printer"
50 PRINT"Druk op een toets en start"
60 PRINT'vervolgens de printerzelftest"
70 REM opnemen van het beeld
80 SCREEN 3:COLOR 1,1,1:CLS
90 DIM C(63,47)
100 IF STRIG(1) THEN 110 ELSE 100
110 FOR Y=0 TO 47
120 FOR X=12 TO 50
130 K=STICK(1)
140 IF K=3 THEN C(X,Y)=12
```

```
150      IF K=5 THEN C(X,Y)=2
160      IF K=7 THEN C(X,Y)=3
170     PSET(4*X,4*Y),C(X,Y)
180      FOR Q=l TO 6:NEXT Q
190    NEXT X
200    IF STRIG(l) THEN 200 ELSE 210
210    FOR Q=1 TO 100:NEXT Q
220    IF STRIG(1) THEN 230 ELSE 220
230 NEXT Y
240 REM menu
250 SCREEN 0:COLOR 1,10,10:CLS
260 PRINT:PRINT:PRINT
270 PRINT" 1 Afbeelden op scherm"
280 PRINT" 2 Beeldpunten opslaan"
290 PRINT" 3 Dump op 3,5 inch-schijf"
300 PRINT" 4 Einde programma"
310 PRINT:PRINT:PRINT
320 PRINT"        Kies nummer";
330 A$=INPUT$(1):A=VAL(A$)
340 IF A<1 OR A>4 THEN 300
350 IF A=1 THEN 350
360 IF A=2 THEN 440
370 IF A=3 THEN 540
380 IF A=4 THEN CLS:KEY ON:END
390 REM afbeelden op scherm
400 SCREEN 3:COLOR 1,1,1:CLS
410 FOR Y=0 TO 47
420   FOR X=0 TO 63
430    PSET(4*X,4*Y),C(X,Y)
440   NEXT X
450 NEXT Y
460 A$=INPUT$(1)
470 IF A$=<>" " THEN 460
480 IF A=3 THEN 640 ELSE 250
490 REM beeldpunten opslaan
500 CLS:FILES
510 INPUT "Naam voor beeldpunten";N$
520 OPEN N$+".PIX" FOR OUTPUT AS #1
530 FOR Y=0 TO 47
540   FOR X=0 TO 63
550    PRINT #!,C(X,Y)
560   NEXT X
570 NEXT Y
580 CLOSE
590 GOTO 250
600 REM dump op 3,5 inch-schijf
610 CLS:FILES
620 INPUT "Geef naam voor scherm-dump op";N$
630 GOTO 390
640 BSAVE N$+".SCR",0,16383,S
650 GOTO 250
```

Het programma is in een aantal blokken verdeeld. De eerste regels geven instructie over het gebruik. Het tweede blok is het eigenlijke opnameprogramma. De te gebruiken kleuren worden vastgelegd in de regels 130 tot en met 150. Het is misschien een aardige BASIC-oefening om dit programma zodanig aan te passen dat de te gebruiken kleuren in het eerste programmablok kunnen worden ingevoerd. Na het opnemen meldt de computer zich met een menu waarin u kunt kiezen uit een aantal mogelijkheden. De laatste drie blokken zorgen ervoor dat wat u uit het menu kiest ook werkelijk gebeurt. Let er goed op dat de vermenigvuldiging van X en Y met een factor 4 alleen maar noodzakelijk is als de beeldpunten op het scherm worden gezet. De factor ontbreekt bij het opslaan van de beeldpunten, omdat die daar overbodig is. Nog een laatste opmerking. Om het beeld netjes in het midden van het scherm te krijgen, loopt de X-coördinaat tijdens de opname en de reconstructie van het beeld op het scherm niet van O tot en met 63. Als we de beeldpunten opslaan, doen we dat echter wél van O tot en met 63. De niet-gekleurde punten krijgen automatisch de kleurwaarde O (transparant). In een later stadium gaan we ook de nog niet door het beeld bezette zwarte marges vullen. Vandaar dat alle X-coördinaten worden opgeslagen.
Zo, dat zit erop. Hard- en software zijn behandeld en het is nu zaak om wat beelden te reproduceren. Wat dacht u van de portretten van uw familieleden, vrienden of kennissen? Gebruik contrastrijke beelden. Al eerder is vermeld dat sommige inktsoorten, zoals viltstiftinkt, slecht infrarood licht absorberen. Dat hoeft geen belemmering te zijn om ze toch door de computer te laten opnemen. U maakt dan namelijk eerst een fotokopie van de prent. Dat is in veel gevallen toch noodzakelijk, omdat papiervorm en -dikte printervriendelijk moeten zijn. Moderne kopieerapparaten bieden bovendien de mogelijkheid om afbeeldingen te vergroten of te verkleinen, waardoor de afmetingen naar wens kunnen worden aangepast. Een kopie op normaal A4-papier kan met een plakbandje aan boven- en zijkanten op het kettingpapier worden vastgezet. In het volgende hoofdstuk wordt het pas echt leuk. We gaan onderzoeken in hoeverre het beeld verder bewerkt en veranderd kan worden.

26 Beeldbewerking

In de voorafgaande hoofdstukken hebben we de beelden opgebouwd in rood-of groenschakeringen.
Misschien heeft u zelf al met andere kleuren geëxperimenteerd. Omdat alle beeldpunten afzonderlijk kunnen worden opgeslagen, zijn we in staat om op elk beeldpunt bewerkingen los te laten. We kunnen dat echter ook op groepen van beeldpunten doen. Bijvoorbeeld op alle beeldpunten met het kleurnum-mer 6, of op alle beeldpunten waarvan de som van de X- en de Y-coördinaat gelijk is aan een bepaald getal. Het totale beeld kan worden verschoven door alle afzonderlijke beeldpunten te verplaatsen. Maar ook kunnen delen van het beeld worden verplaatst. Ook kunnen de punten met eenzelfde kleur als afzonderlijke beelden worden opgeslagen.
Plak een afbeelding met de bovenrand vast op het papier van de printer. Plaats aan de zijkanten ook een stukje plakband. We zijn er dan van verzekerd dat de afbeelding goed door het printerpapier wordt meegenomen. Als uw printer is uitgerust met een wrijvingsrol, in plaats van een pinfeed-mechanisme, dan zal het duidelijk zijn dat alle opmerkingen in dit boek over het opplakken van afbeeldingen niet voor u zijn bestemd.
Start het opnameprogramma en breng het beeld over op het scherm. Afhankelijk van de aard van het over te nemen beeld, zouden de drie kleurnuances ongeveer een gelijke oppervlakte moeten beslaan. Is dat niet het geval, dan is dit een mooie gelegenheid om de twee instelpotentiometertjes nog wat af te regelen. Een verdraaiing linksom laat het desbetreffende niveau zakken. Kies uit het menu de optie waarbij alle beeldpunten worden opgeslagen en geef het bestand de naam 'COPIE'.
Ga na of op de band of de diskette nu het bestand COPIE.DAT voorkomt. Zo ja, dan is alles gelukt en voeren we na de opdracht NEW, die het geheugen ontruimt, het volgende programma in.

```
10  REM kleurverandering
20  DIM C(63,47)
30  SCREEN 0:COLOR 1,10,10:CLS:KEY OFF
40  FILES
50  INPUT "Naam ";N$
60  OPEN N$+".PIX"
70  SCREEN 3:COLOR 1,1,1:CLS
80  FOR Y=0 TO 47
90   FOR X=0 TO 63
100     INPUT #1,C(X,Y)
110     IF C(X,Y)=12 THEN C(X,Y)=4
120     IF C(X,Y)=3 THEN C(X,Y)=10
130    PSET(4*X,4*Y),C(X,Y)
140   NEXT X
150 NEXT Y 160 CLOSE 170 GOTO 170
```

Het resultaat is sprookjesachtig. De kleuren waarin het beeld opgenomen is, zijn naar keuze te wijzigen. In regel 110 en 120 kunnen we elke kleurtoekennen aan twee van de drie oorspronkelijke kleuren. Door de toevoeging van nóg een regel kan ook de derde kleur naar wens worden veranderd.

```
122   IF C(X,Y)=2  THEN C(X,Y)=14
```

Het afgedrukte programma leent zich uitstekend voor variaties. We zullen het deel van het programma dat het beeld van het opslagmedium haalt apart zetten en er een blokje aan toevoegen waarmee het nieuwe beeld gesaved kan worden. Daarbinnen wordt ook een plaatsje ingeruimd voor de beeldbewerking zelf.

```
10 REM beeldbewerking
20 DIM C(63,47)
30 SCREEN 0:COLOR 1,10,10:CLS:KEY OFF
40 FILES
50 INPUT "Naam ";N$
60 OPEN N$+".PIX" FOR INPUT AS #1
70 SCREEN 3:COLOR 1,1,1:CLS
80 FOR Y=0 TO 47
90 FOR X=0 TO 63
100 INPUT #1,C(X,Y)
130 PSET(4*X,4*Y),C(X,Y)
140 NEXT X
150 NEXT Y
160 CLOSE
200 GOSUB 1000
300 SCREEN 1:COLOR 1,10,10
310 FILES
320 INPUT "Naam ";N$
330 OPEN N$+".PIX" FOR OUTPUT AS #1
340 FOR Y=0 TO 47
350   FOR X=0 TO 63
360    PRINT #1,C(X,Y)
370  NEXT X
380 NEXT Y
390 CLOSE
```

De regels 110 en 120 zijn verdwenen. Regel 40 (alleen voor gebruikers van een 3,5 inch-schijfeenheid) zorgt ervoor dat alle bestanden die op de diskette staan, worden afgebeeld op het scherm. Een dubbel naamgebruik, met het daarbij behorende gevolg (oude bestand wordt overschreven), kan hierdoor worden vermeden. Let er bij het invoeren van een bestandsnaam wel op dat die zonder de toevoeging .DAT moet worden ingetypt.

De hierna afgedrukte subroutine die aan het hoofdprogramma kan worden toegevoegd, maakt het mogelijk een kleur te isoleren. Een nieuw beeld, waarin slechts één van de kleurschakeringen is geselecteerd, wordt opgeslagen. Een mooie gelegenheid om ook de kleur even aan te passen.

```
1000 REM solokleur
1000 FOR Y=0 TO 47
1010 FOR X=0 TO 63
1020 INPUT #1,C(X,Y)
1030 IF C(X,Y)=12 THEN C(X,Y)=4
1040 IF C(X,Y)=3 THEN C(X,Y)=1
1050 IF C(X,Y)=2 THEN C(X,Y)=1
1060 PSET(4*X,4*Y),C(X,Y)
1070 NEXT X
1080 NEXT Y
1090 IF STRIG(0) THEN RETURN ELSE 1080
```

In dit geval is donkergroen vervangen door blauw. Groen en lichtgroen zijn verdwenen, omdat zwart tegen een zwarte achtergrond niet zichtbaar is. Het subroutineblok laat zich op vele manieren samenstellen. Zo kan het tijdens het opnemen van een afbeelding gebeuren dat bepaalde punten in het gereproduceerde beeld ons toch niet zo goed bevallen. Bij een portret kan een lichtpuntje in de ogen wonderen doen. Het volgende blok geeft ons de mogelijkheid ieder afzonderlijk punt te beïnvloeden. We maken gebruik van de cursorbesturing op het toetsenbord, in combinatie met de spatiebalk.

```
1000 REM pixel verandering
1010 XO=0:YO=0:X=0:Y=0
1020 A=STICK(0)
1030 X=XO-(A=2)-(A=3)-(A=4)+(A=6)+(A=7)+(A=8)
1040 Y=YO+(A=1)+(A=2)+(A=8)-(A=4)-(A=5)-(A=6)
1050 PSET(4*X,4*Y),15
1060 FOR Q=l TO 10:NEXT Q
1070 PSET(4*X,4*Y),1
1080 FOR Q=l TO 10:NEXT Q
1090 IF STRIG(0) THEN GOSUB 1200
1100 IF X=XO AND Y=YO THEN 1020
1110 PSET(XO,YO),C(XO,YO)
1120 XO=X:YO=Y
1130 GOTO 1020
1200 REM subroutine kleurverandering
1210 A=STICK(0)
1220 IF A=1 THEN 300
1230 T=T+(A=7)-(A=3)
1240 IF T<1 THEN T=l
1250 IF T>15 THEN T=15
1260 C(X,Y)=T
1270 PSET(4*X,4*Y),C(X,Y)
1280 IF STRIG(O) THEN RETURN ELSE 1210
```

Nadat het eerste programmadeel (tot regel 170) het oorspronkelijke beeld op het scherm heeft gezet, verschijnt linksboven een knipperende cursor in beeld (regel 1050-1080).
Met de cursortoetsen kan het knipperlichtje naar elke plaats op het scherm worden bewogen. Zodra het naar een punt is gedirigeerd waarvan de kleur ons niet bevalt, drukken we op de spatiebalk (regel 1090). De subroutine vanaf regel 1200 geeft de mogelijkheid om de kleur van het desbetreffende punt te wijzigen met behulp van de horizontale cursorbesturings-toetsen (regel 1230). Druk hierbij niet per ongeluk op de 'omhoog-toets'. Zodra de kleur naar tevredenheid is, brengt een tweede druk op de spatiebalk (regel 1280) ons terug in het hoofdprogramma. Na de correctie van het laatste beeldpunt, drukken we op de 'omhoog-toets' en het nieuwe beeld wordt opgeslagen. De cassetterecordergebruikers weten intussen welke veranderingen cq. toevoegingen nodig zijn om het programma voor hen geschikt te maken. Met dit programma kunnen ook punten buiten het opgenomen beeld worden veranderd. Zo kunnen de linker- en rechtermarge worden ingekleurd. Echter, als deze gebieden overal dezelfde kleur moeten hebben, is dat sneller te realiseren door de achtergrondkleur in regel 70 aan te passen. Voor de echte programmeur is dit programma een voorbeeld van minderfraai programmeerwerk. Het programma komt in een subroutine terecht, waar het niet meer uitkomt met een RETURN-opdracht, maar met een GOTO-opdracht. In dit geval verstoort het de loop van het programma echter niet, omdat na het opslaan van de gegevens het werk is gedaan.
Ter afsluiting van dit hoofdstuk bekijken we nog een techniek die in een later stadium van pas zal komen. Het programma is weer als subroutine uitgevoerd. Reclame-ontwerpers gebruiken de computer intensief. Eén van de methodes die zij gebruiken, is het omzetten van het beeld in horizontale lijnen met variërende dikte. Op het beeldscherm kunnen we dat enigszins nabootsen door de grafische karakters te gebruiken die met de toetsen u,i,o en p kunnen worden opgeroepen. Het beeld wordt dan niet in drie kleurschakeringen uitgevoerd maar de grootte van het blokje bepaalt de gemiddelde helderheid.

```
1000 REM lijndikte
1000 FOR Y=0 TO 47
1010   FOR X=0 TO 63
1020     INPUT #1,C(X,Y)
1030     IF C(X,Y)=12 THEN PRINT"■ ";
1040     IF C(X,Y)=3 THEN PRINT"▬ ";
1050     IF C(X,Y)=2 THEN PRINT"▂ ";
1060     IF C(X,Y)=1 THEN PRINT"█ ";
1070   NEXT X
1080   PRINT" "
1090 NEXT Y
1100 IF STRIG(0) THEN RETURN ELSE 1100
```

Regel W30: [GRAPH][o] Regel 1040: [GRAPH][i] Regel 7050: [GRAPH][u] Regel 1060: [GRAPH][p]

In hoofdstuk 28, dat het afdrukken van bewerkte beelden behandelt, wordt een methode beschreven die hier erg veel op lijkt en die kan resulteren in een ongekend fraaie uitvoer op papier. We hebben nu een aantal bewerkingen mogelijk gemaakt die gebaseerd zijn op het manipuleren van kleuren. Het volgende hoofdstuk laat zien dat er met het verplaatsen van punten ook verrassende effecten te bereiken zijn.

27 Geometrische beeldbewerking

In het vorige hoofdstuk zijn enkele gereedschappen ontwikkeld, waarmee de kleuren van een opgenomen beeld kunnen worden gewijzigd. In dit hoofdstuk gaan we hetzelfde doen voor plaatsveranderingen van beeldpunten. Het eerste deel van het programma uit het vorige hoofdstuk blijft ongewijzigd. Hiermee (regel 10 tot 170) wordt een opgeslagen beeld op het scherm gezet. De beeldbewerking verrichten we steeds in een aanvullend programmadeel, dat wordt toegevoegd aan het laadprogramma.
De afbeelding wordt tijdens de opname ervan gecentreerd op het beeldscherm geplaatst. Als de opname echter binnen andere programma's gaat worden gebruikt, is het helemaal niet gezegd dat die centrale plaats ook de juiste is. We beginnen dan ook met een verschuiving van het beeld in horizontale richting.



Afbeelding 27-1 Uw auteur opgebouwd uit pixels. Voor een goed beeld van de pixel-plaatjes dient u het boek wat van u af te houden.

```
170 REM horizontale verschuiving
180 FOR Y=0 TO 47
190   FOR X=0 TO 63
200    XN=X-5
210    PSET(4*XN,4*Y),C(X,Y)
220   NEXT X
230 NEXT Y
240 GOTO 240
```

De verschuiving van het beeld langs de X-as blijkt niet zo moeilijk te zijn. In regel 200 wordt een nieuwe horizontale coördinaat XN berekend. In dit geval treedt er een verschuiving op van vijf beeldpunten in de negatieve richting,

naar links dus. In de PSET-opdracht wordt de kleur, die eigenlijk bij punt X behoort, afgebeeld op de nieuwe plaats XN. Het programma verricht deze operatie op alle beeldpunten van het scherm (regel 180 en 190). Als we in regel 200 XN=X+32 zetten, wordt het beeld aan de rechterzijde (gedeeltelijk) van het scherm geduwd.
Dezelfde methode is ook toe te passen voor een verticale verschuiving van de afbeelding. In dat geval berekenen we een nieuwe Y-coördinaat.

```
170 REM verticale verschuiving
180 FOR Y=0 TO 47
190 YN=Y-5
200 FOR X=0 TO 63
210 PSET(4*X,4*YN),C(X,Y)
220 NEXT X
230 NEXT Y
240 GOTO 240
```

Om te voorkomen dat de nieuwe Y-coördinaat bij elke X-waarde opnieuw moet worden berekend, zijn - in vergelijking tot het vorige programma - de regels 190 en 200 omgewisseld. Nu wordt in regel 190 de nieuwe Y-coördinaat (YN) berekend. Als dit programma als aanvulling op het hoofdprogramma uit het vorige hoofdstuk wordt uitgevoerd, blijkt dat er toch iets merkwaardigs aan de hand is. Het beeld wordt weliswaar over een afstand van vijf beeldpunten (regel 190) naar boven verschoven, maar de onderste vijf rijen van het oude beeld blijven staan. Dit euvel kunnen we gemakkelijk verhelpen door de FOR-NEXT-lus die de Y-waarden doorloopt, aan de nieuwe situatie aan te passen.

```
170 REM verticale verschuiving
180 FOR Y=0+5 TO 47+5
190  YN=Y-5
200   FOR X=0 TO 63
210     PSET(4*X,4*YN),C(X,Y)
220   NEXT X
230 NEXT Y
240 GOTO 240
```

Voor de duidelijkheid zijn de nieuwe Y-waarden in regel 180 niet uitgerekend. Door het op deze manier te schrijven, blijft de Y-verplaatsing herkenbaar. Het kan nog iets netter worden opgeschreven, zodat we de gewenste verschuiving maar op één plaats in hoeven te voeren.

```
170 REM verticale verschuiving
180 V=5
190 FOR Y=0+V TO 47+V
200  YN=Y-V
210   FOR X=0 TO 63
220    PSET(4*X,4*YN),C(X,Y)
230   NEXT X
240 NEXT Y
250 GOTO 250
```

Het beeldscherm wordt nooit helemaal door het opgenomen prentje gevuld. Toch is dat in veel gevallen wel gewenst. Natuurlijk kunnen we het beeld horizontaal wat uitrekken. De onderlinge verhoudingen gaan dan echter verloren en in veel gevallen is dat niet acceptabel. Omdat alle beeldinformatie beschikbaar is en het beeld bovendien op elke plaats kan worden neergezet, is het geschetste probleem voor een deel te omzeilen. We passen eenvoudigweg beeldverdubbeling toe. Het beeld wordt tweemaal naast elkaar geplaatst, en wel zodanig dat beide beelden de schermbreedte volledig benutten. Vooral bij beelden met een horizon, maar ook bij geometrische motieven, kan dit alleraardigste effecten opleveren.





Afbeelding 27-2 Een portret van Kobus en een portret van Kobus met zijn tweelingbroer: beeldverdubbeling.

```
170 REM dubbelganger
180 FOR Y=0 TO 47
```

```
190  FOR X=0 TO 63
200    PSET(4*(X-12),4*Y),C(X,Y)
210    PSET(4*(X+31),4*Y),C(X+12,Y)
220  NEXT X
230 NEXT Y
240 GOTO 240
```

Als we het beeld kunnen verdubbelen, dan mag spiegelen ook geen grote problemen opleveren.

```
170 REM spiegel 180 FOR Y=0 TO 47 190  FOR X=0 TO 63
200    PSET(4*(X-12),4*Y),C(X,Y)
210    IF X>31 THEN 240
220    PSET(4*94-4*(X+31),4*YN),C(X+12,Y)
230   NEXT X
240 NEXT Y
250 GOTO 250
```

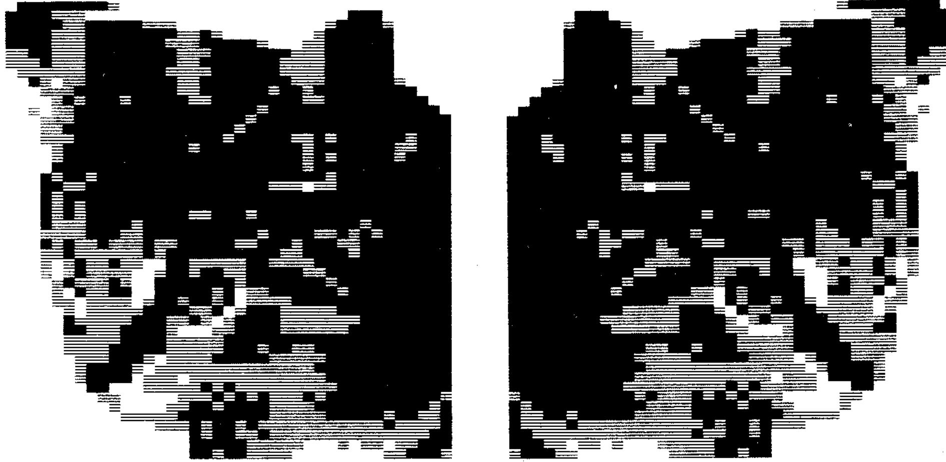


Afbeelding 27-3 De auteur kijkt zichzelf diep in de ogen: beeldspiegeling.

Regel 210 zorgt ervoor dat het gespiegelde rechterbeeld niet over het linkerbeeld wordt geschreven. In regel 220 laten wedeY-coördinaat van rechts naar links lopen. In het getal 94, dat voor die omkering zorgt, zit bovendien de rechter marge verrekend. De rechtermarge valt op deze wijze keurig buiten het beeld.
Voor het spiegelen van beelden komt er, zoals blijkt, wat eenvoudig hoofdrekenwerk om de hoek kijken. Veel simpeler is het omdraaien van het beeld. Dus links-rechts verwisseld.

```
170 REM links=rechts
180 FOR Y=0 TO 47
190 FOR X=0 TO 63
```

```
200 PSET(4*(62-X),4*Y),C(X,Y)
210 NEXT X
220 NEXT Y
230 GOTO 230
```



Afbeelding 27-4 Deze twee poezen willen niets meer met elkaar te maken hebben: beeldomkering.

Hiervoor hoeft de X-coördinaat slechts van rechts naar links, in plaats van van links naar rechts te lopen. We trekken de X-coördinaat telkens van de maximale X-waarde af. Voor spiegeling in de as die horizontaal halverwege het beeldscherm loopt, geldt iets dergelijks.

```
170 REM onder=boven
180 FOR Y=0 TO 47
190   FOR X=0 TO 63
200    PSET(4*X,4*(47-Y)),C(X,Y)
210   NEXT X
220 NEXT Y
230 GOTO 230
```

De beeldopbouw begint nu onderaan het scherm in plaats van bovenaan. Het op deze wijze gevormde beeld is een gespiegeld beeld, dat op z'n kop staat. Om het beeld ongespiegeld ondersteboven op het scherm te zetten, kunnen we de programmadelen 'links=rechts' en 'onder=boven' combineren.

```
170 REM ondersteboven
180 FOR Y=0 TO 47
190  FOR X=0 TO 63
200    PSET(4*(62-)X,4*(47-Y)),C(X,Y)
210   NEXT X
220 NEXT Y
230 GOTO 230
```

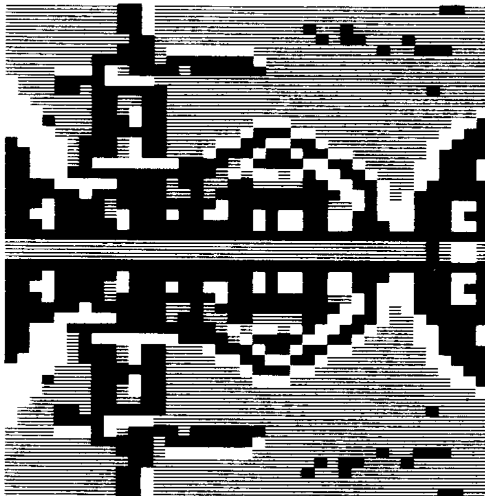


Afbeelding 27-5 Een drukfout? Nee, het beeld staat op de kop door een dubbele beeld-omkering.

Als we nu het TV-toestel of de monitor op z'n kop zetten hebben we het oude beeld terug.

Nog twee beeldgrapjes tot besluit.

```
170 REM waterspiegel
180 FOR Y=0 TO 23
190   FOR X=0 TO 63
200     PSET(4*X,4*(47-Y)),C(X,Y)
210   NEXT X
220 NEXT Y
230 GOTO 230
```



Afbeelding 27-6 Noordhollands polderlandschap: waterspiegel.

Op het eerste gezicht is dit programmadeel identiek met het programmadeel 'onder=boven'. De tweede regel is echter verschillend. Door de Y-coördinaat maar tot de helft door te laten lopen, wordt de bovenste helft van de afbeelding in de onderste helft gespiegeld weergegeven. Een tafereeltje dat in ons land aan elke slootkant is te bewonderen.

```
170 REM speelkaart
180 FOR Y=0 TO 23
190   FOR X=0 TO 63
200    PSET(4*(62-X),4*(47-Y)),C(X,Y)
210   NEXT X
220 NEXT Y
230 GOTO 230
```



Afbeelding 27-7 Speelkaart: de dame.

Dit programma lijkt op de waterspiegel. In de onderste helft van het beeld wordt het gespiegelde beeld echter ook in de X-richting omgekeerd. Het prentje vertoont daardoor sterke gelijkenis met een speelkaart. Het ziet er naar uit dat we de beeldopbouw aardig in de vingers beginnen te krijgen. We kunnen zelfs volslagen onzinnige operaties op de beeldpunten uitvoeren. Maar ook het coderen van een beeld behoort tot de mogelijkheden. Via een bepaalde sleutel kan het beeld worden vervormd. Alleen als de code bekend is, kan de reconstructie van het beeld plaatsvinden. Er zijn mogelijkheden te over, maar... genoeg gespeeld.
Het wordt tijd om de opgedane beeldbewerkingservaring in een gemakkelijk hanteerbaar programma te vatten. In een volgend hoofdstuk wordt de nodige aandacht aan zo'n meesterwerk besteed. Maar eerst bekijken we de mogelijkheden om een afdruk van het gedigitaliseerde beeld op papier te maken.

28 Uitvoer op de printer

MSX-computers zijn niet standaard voorzien van een screen dump-functie. Met zo'n functie kan met een bepaalde toetscombinatie de scherminhoud worden overgebracht op papier. In de literatuur zullen ongetwijfeld programma's te vinden zijn die in staat zijn deze functie te verrichten. We zullen in dit boek geen screen dump-routine ontwikkelen. We maken gebruik van een methode die op alle matrix-printers een bevredigend resultaat zal geven, maar die ook op een margriet(daisy)wiel-printer bruikbaar is. Over de beeldkwaliteit van het laatste printertype moeten we ons niet al te veel illusies maken,omdat deze machines uitsluitend zijn bedoeld voor het weergeven van tekst. Toch is het beter dan niets. Als we van plan zijn een behoorlijk aantal prentjes op te nemen, kunnen we de printerafdrukken uitstekend gebruiken om een catalogus van die beelden samen te stellen.
Het omzetten van een pixelkleur naar een letter levert een redelijk resultaat op. De donkerste toon kunnen we op papier laten representeren door een X en de middentoon doorx. Er ontstaat op deze wijze een zekere helderheidsvariatie.

```
IF   C(X,Y)=12   THEN  LPRINT   "X";
IF  C(X,Y)=2   THEN  LPRINT   "x";
IF  C(X,Y)=3   THEN  LPRINT   "-";
IF  C(X,Y)=1   THEN  LPRINT   "   ";
```

Een streepje neemt de rol van de lichtste kleur op zich, terwijl spaties de achtergrond weergeven. De puntkomma zorgt ervoor dat de tekens direct achter elkaar worden afgedrukt. Een printprogramma zou er dan zó uit kunnen zien:

```
10 REM print
20 DIM C(63,47)
30 SCREEN 0:COLOR 1,10,10:CLS:KEY OFF
40 FILES
50 INPUT"Naam";N$
60 OPEN N$+".PIX" FOR INPUT AS #1
70 FOR Y=0 TO 47
80   FOR X=0 TO 63
90     INPUT #1,C(X,Y)
100     IF C(X,Y)=12 THEN LPRINT "X";
110     IF C(X,Y)=2 THEN LPRINT "x";
120     IF C(X,Y)=3 THEN LPRINT "-";
130     IF C(X,Y)=1 THEN LPRINT " ";
140   NEXT X
150   LPRINT
160 NEXT Y
170 CLOSE
```

In programmaregel 150 wordt het papier naar de volgende regel gedraaid. De uitvoer van dit programma ziet er een beetje vreemd uit. We hebben in het beeldopnameprogramma het beeld gecomprimeerd in horizontale richting. Dat moeten we nu eigenlijk weer compenseren. Door het aanbrengen van een kleine aanpassing in de regels 100 tot en met 130, bereiken we dat het beeld op papier er wat normaler uitziet.

```
100 IF C(X,Y)=12 THEN LPRINT "XX";
110 IF C(X,Y)=2 THEN LPRINT "xx";
120 IF C(X,Y)=3 THEN LPRINT "--";
130 IF C(X,Y)=1 THEN LPRINT "  ";
```

Alle tekens worden verdubbeld, waardoor er een resultaat ontstaat dat voor archiefdoeleinden bruikbaar is. Voor het op deze manier afdrukken van de afbeelding moet de printer wel 2x64= 128 tekens opeen papierregel kwijt kunnen. In het algemeen hebben de printers die bij homecomputers worden gebruikt slechts 80 kolommen. Als het plaatje niet de gehele breedte van het beeldscherm bestrijkt, kunnen we de X-coördinaat over slechts een gedeelte van de X-as laten lopen. We deden dat ook al in het beeldopnameprogramma.

```
80   FOR X=12  TO  50
```

Een beter alternatief wordt gevormd door, als dat tenminste mogelijk is, enkeletekens ('X', 'x'en '-') te gebruiken en de regelhoogte van de printer te halveren.

De matrixprinter

Vrijwel alle matrixprinters kennen een beperkte hoeveelheid grafische tekens. Door hiervan gebruikte maken, kan de kwaliteitvan de afdruk iets worden verbeterd. De grafische tekens kunt u in de printerhandleiding opzoeken. Kies een teken meteen donkere'toon' uit. Een volledig zwart vlak voldoet nog het beste aan onze wens. Het teken moet wel de hele ruimte die er voor een teken beschikbaar is, opvullen, zodat er aan weerszijden van het afgedrukte teken geen witte randjes overblijven. Voor de middentint zoeken we een ander teken uit: een teken met een gemiddelde grijstint. De spatie dient als lichtste kleur. Voor de achtergrond kunt u een geschikt teken uitzoeken, maar u kunt ook hiervoor een spatie gebruiken. In de handleiding van de printer staat welke decimale codes bij de tekens horen. Is dat niet het geval, dan staat er in veel gevallen een hexadecimale code.
Voor MSX-printers zijn de tekens met de decimale codes 215 en 203 redelijk geschikt voor het weergeven van respectievelijk de donkerste 'toon' en de midden-'toon'. De spatie die de lichtste kleurschakering representeert, heeft het getal 255 als decimale code. Voor de achtergrond zou het teken met code 248 kunnen worden toegepast.
We zullen een ander voorbeeld bekijken aan de hand van een goedkope ther-

mische printer: de Brother HR-5. Als speciale grafische tekens zijn de tekens met de hexadecimale codes 8E en FE bij uitstek geschikt om als donkerste 'toon', respectievelijk midden-'toon' te dienen. De spatie heeft het getal 20 als hexadecimale code, terwijl de achtergrond opgevuld kan worden door tekens met de code 8F. Het afdrukken van een teken met een decimale karaktercode gaat als volgt:

```
LPRINT  CHR$(249);
```

De puntkomma zorgt ervoor dat de printer het volgende teken er direct achter plaatst. Voor het afdrukken van een teken met een hexadecimale karaktercode moeten we het voorvoegsel '&H' aan de code toevoegen.

```
LPRINT(&H8E) ;
```

Het printen op een MSX-printer gaat nu als volgt:
Breng het beeld vanaf het opslagmedium in het geheugen met behulp van het beeldweergaveprogramma, waarin de regels 100 tot en met 130 worden vervangen door:

```
100 IF C(X,Y)=12 THEN LPRINT CHR$(215);
110 IF C(X,Y)=2 THEN LPRINT CHR$(203);
120 IF C(X,Y)=3 THEN LPRINT CHR$(255);
130 IF C(X,Y)=1 THEN LPRINT CHR$(248);
```

Om de hoogte-breedte-verhouding van het beeld in de juiste verhouding op papier te krijgen, moet de printer op halve regelafstand worden gezet. Voor een doorsnee-matrixprinter, moeten de regels 100 tot en met 130 er zó uit komen te zien:

```
100 IF C(X,Y)=12 THEN LPRINT CHR$(&H8E);
110 IF C(X,Y)=2 THEN LPRINT CHR$(&HFE);
120 IF C(X,Y)=3 THEN LPRINT CHR$(&H20);
130 IF C(X,Y)=1 THEN LPRINT CHR$(&H8F);
```

Ook hier moet worden geprint met halve regelafstand. Kijk in uw printerhandleiding hoe de regelafstand gehalveerd kan worden.
Een veel beter resultaat kan worden bereikt door zelf tekens te ontwerpen die de gewenste 'toon' hebben. Bovendien kunt u ze dan op elkaar laten aansluiten. Hiervoor dient de printer geschikt te zijn voor het afbeelden van bitpatronen. Is dat het geval dan vermeldt de printerhandleiding hoe het definiëren van een teken in z'n werk gaat. De in dit boek afgedrukte pixelbeelden bestaan uit van die zelf-gedefinieerde tekens.
Een aardige variant, die bij de beeldbewerking met kleuren al aan de orde is geweest, mag hier niet achterwege blijven. Door het beeld niet af te drukken in grijstinten, maar in lijndikten, ontstaat het effect dat in de reclamewereld veel

wordt toegepast. Als printtekens worden dan lijntjes gebruikt met oplopende dikte.

Het oplossend vermogen van screen 3 van de MSX-computer is prachtig om er beelden mee vast te leggen ter ondersteuning van activiteiten op het scherm. Gedrukt op papier met een gewone printer gaat er veel van de charme van het uit pixels opgebouwde beeld verloren. De hierboven behandelde printmethoden dienen dan ook slechts beschouwd te worden als archivarisch hulpmiddel.
Een heel ander verhaal wordt het als we voor het afdrukken gebruikmaken van een plotter en de methode waarbij lijntjes met verschillende diktes worden gebruikt.
De kwaliteit van het beeld is zonder meer vergelijkbaar met de beeldschermuitvoering. Eigenlijk staat het op papier extra leuk.
Het volgende piotprogramma is ontworpen voor de zeer laag geprijsde MSX-printer/plotter Toshiba HX-P570, die in dezelfde uitvoering ook onder een andere merknaam in de winkel kan worden aangetroffen.

```
10 REM plotterafdruk
20 DIM C(63,47)
30 SCREEN 0:COLOR 1,10,10:CLS:KEY OFF
40 FILES
50 INPUT "Naam";N$
60 LPRINT "Afdruk van ";N$
70 LPRINT CHR$(&HlB)+"#"
80 OPEN N$+".PIX" FOR INPUT AS #1
90 FOR Y=0 TO 47
100   FOR X=0 TO 63
110     INPUT #l,C(X,Y)
120     IF C(X,Y)=12 THEN LPRINT "J4,0,0,1,-4,0,0,
           1,4,0":LPRINT"RO,-2"
130     IF C(X,Y)=2 THEN LPRINT "J4,0,0,l,-4,0":
           LPRINT"R0,-1"
140 IF C(X,Y)=3 THEN LPRINT "J4,0"
150 IF C(X,Y)=1 THEN LPRINT "R4,0"
160 IF C(X,Y)=0 THEN LPRINT "R4,0"
170 NEXT X
180   LPRINT "F":LPRINT "R0,17" 190 NEXT Y 200 CLOSE
```

De lijntjes worden getekend in de regels 120 tot en met 140 en hebben een lengte van vier beeldpunten. Het dikste lijntje is drie beeldpunten dik, het middenmaatje is twee beeldpunten dik en het dunste lijntje één. De beeldpuntafstand op de plotter is 0,2 mm. Dat heeft tot gevolg dat de totale afbeelding ongeveer 5x3 cm wordt. Een mooie afmeting om uw briefpapier een zeer persoonlijk cachet te geven. Wilt u een grotere afbeelding? Dan kunt u de diktes en de lengtes van de lijnen in de regels 120 tot en met 140 aanpassen. Houd er wel rekening mee dat de plotter er lang over doet om een grote afbeelding op papier te tekenen.

29 Iets over het geheugen

Vóór het beeldbewerkingsprogramma wordt opgebouwd, moeten we even stilstaan bij de geheugenruimte die voor onze experimenten beschikbaar is. Om hiervan een goed beeld te krijgen, is het nodig om te weten hoeveel geheugenruimte een compleet beeld ongeveer vergt.
Het/multicolor-scherm bestaat uit 48 regels. Op elke regel is plaats voor 64 beeldpunten. In totaal moet de computer 3072 beeldpunten op kunnen slaan. Elk beeldpunt heeft twee coördinaten en een kleur. Doordat we in dit boek de drie getallen samenvatten in een tweedimensionaal-array-element C(X,Y), slaan we per beeldpunt maar één getal op.
Een array-element neemt zonder verdere maatregelen acht bytes van het geheugen in beslag. Een beeld wordt dus beschreven door 3072 getallen die elk acht bytes in beslag nemen. Van het geheugen zouden dan ook 3072 x 8=24576 bytes moeten worden gereserveerd voor de opslag van een beeld. We hebben dus 24576/1024= 24 Kb (kilo-byte) nodig voor de opslag van een beeld. Het vrij beschikbare geheugen voor een 64K-machine is ongeveer 28 Kb. We hebben dan voor een programma slechts 4 Kb over. Dat is wel erg weinig als we in dat programma alle beeldmanipulatie-mogelijkheden die we al kennen, willen inbouwen.
In MSX-BASIC kunnen we variabelen ook in een ander jasje steken:

Gehele getallen	: 2 bytes per getal
Getallen met enkele precisie	: 4 bytes per getal
Getallen met dubbele precisie	: 8 bytes per getal

We zullen aan de hand van een miniprogramma demonstreren hoeveel geheugen voor een beeld nodig is als we de door ons gebruikte array-elementen onderbrengen in de verschillende categorieën. Als we geen maatregelen nemen, slaat de MSX-computer alle variabelen op met dubbele precisie (14 cijfers per getal).
Zodra we met de DIM-opdracht ruimte voor de opslag van een beeld reserveren, kunnen we uitrekenen hoeveel geheugen er nog voor het programma overblijft. De computer kan dat zelf ook.
Na het aanzetten van de computer verschijnt op het scherm een mededeling over de grootte van het beschikbare geheugen:

```
MSX BASIC version 1.0
Copyright 1983 by Microsoft
28815 Bytes free
```

We typen nu een regel in, waarin we de omvang van de array C(X,Y) vastleggen en bovendien laten we de voor het programma en andere gegevens beschikbare geheugenruimte afdrukken:

```
DIM C(64,48):PRINT FRE(l)    [Return]
4208
```

Ongeveer 4 Kb, dat is niet veel voor een beetje uitgebreid programma. We kijken nu hoe het verloopt als we de array C(X,Y) met enkele precisie op laten slaan. Dat kunnen we aangeven door direct achter de variabelenaam een uitroepteken te plaatsen:

```
DIM C!(64,48):PRINT  FRE(l)    [Return]
16495
```

Er is nu maar liefst ruim 16 Kb beschikbaar, dat scheelt aanzienlijk.
Wat we eigenlijk willen opslaan zijn kleurnummers, C(X,Y) staat immers voor de kleur van het beeldpunt X,Y. Nummers zijn gehele getallen. Als we de array C(X,Y) laten opslaan als een reeks gehele getallen ziet de uitkomst er als volgt uit:

```
DIM C%(64,48):PRINT  FRE(1)
22683
```

Dat begint erop te lijken. De opslag van een beeld vraagt nu nog slechts 4 Kb. We hebben ongeveer 24 Kb over voor onze werkzaamheden. Dat is meer dan voldoende voor een zeer uitgebreid programma.
Anders ligt het bij de opslag op 3,5 inch-schijf. Bij het gebruik van C(X,Y), dus met dubbele precisie, vraagt een compleet beeld 17 Kb aan opslagruimte. Als C%(X,Y) wordt gebruikt, slaan we de kleurinformatie als een rij gehele getallen op. Toch vraagt dit nog altijd 16 Kb. (Dat is precies even veel als we voor de opslag van het videogeheugen nodig hebben.) De consequentie hiervan is dat er maar een beperkt aantal beelden op een schijf passen. Proberen we er per ongeluk te veel op te zetten, dan verschijnt op het beeldscherm de foutmelding: 'Device I/O error'. Laat het schijfje dan even zitten en reageer met het intypen van de CLOSE-opdracht. Daardoor kunnen er met deze schijf geen vreemde dingen gebeuren. Controleer dus voordat u beelden gaat opslaan, of er op de schijf genoeg ruimte is.

30 Beeldbewerking a la carte

De vele mogelijkheden om opgenomen beelden te modificeren, zijn in de vorige hoofdstukken de revue gepasseerd. In dit hoofdstuk bouwen we een menu-gestuurd programma op, waarmee vrijwel alle denkbare functies kunnen worden verricht. We beginnen met een schone lei. In dit geval dus met een lege cassetteband of een lege 3,5 inch-schijf.
Om te voorkomen dat we het programma niet meer kunnen bedienen als we het enige tijd niet hebben gebruikt en we zijn de exacte werking vergeten, maken we er een menu-gestuurd programma van dat aan duidelijkheid niets te wensen overlaat.

```
HOOFDMENU

1 Opnemen
2 Weergeven
3 Veranderen

Kies nummer
```

Vanaf dit hoofdmenu kunnen we een optie kiezen. Bij elke optie hoort een submenu. De inhoud van die submenu's zou er uit kunnen zien zoals hieronder afgebeeld:

```
MENU 1

Opnemen op:

1 Cassette
2 Quickdisk
3 Diskdrive

Kies nummer
```

Achter elke keuzemogelijkheid schuilt een subroutine die zorg draagt voor de juiste verrichting van de opdracht.
Het is onnodig ingewikkeld om het programma geschikt te maken voor het gebruik van alle in aanmerking komende opslagmedia. Iemand met een 3,5 inch-schijfeenheid zal weinig behoefte hebben aan een opslagmogelijkheid op cassette, terwijl bezitters van een zgn. Quickdisk weinig opschieten met een opslagmethode op 3,5 inch-diskette.
Het vorige hoofdstuk heeft aangetoond dat we niet zuinig met het beschikbare

geheugen om hoeven te springen. Toch zullen we terwille van de overzichtelijkheid het programma in vier afzonderlijke delen opsplitsen. Dat werkt gemakkelijk als we later veranderingen aan willen brengen. Bovendien duurt het laden van een programma-onderdeel korter dan het laden van het hele programma.
Om de opbouw van het programma helderte houden, beperken we ons tot het gebruik van de 3,5 inch-diskette als opslagmedium. Gebruikers van cassettebandjes kunnen de programma-onderdelen aanpassen. Dit komt er in het algemeen op neer dat voor een bestandsnaam 'CAS:' moet worden geplaatst. Een programmaregel zoals:

```
10  OPEN   "PLAATJE.PIX"   FOR   INPUT  AS  #1
```

wordt dan:

```
10  OPEN   "CAS:PLAATJE.PIX"   FOR  INPUT  AS  #1
```

Lees eerst uw BASIC-handleiding goed door om na te gaan hoe bestanden op uw opslagmedium opgeslagen c.q. van uw opslagmedium gelezen dienen te worden.
Vanwege de bijzondere kwaliteit van plotteruitvoer zal de in te bouwen print-routine niet alleen geschikt worden gemaakt voor printers (ongeacht het type), maar ook voor de plotter.
Het hoofdprogramma krijgt de naam 'PIXEL'. Dit programma geeft de mogelijkheid te kiezen uit:

- Het opnemen van beelden met de optische lezer die op de printer is gemonteerd.
- Het weergeven van beelden. Zowel beelden die als verzameling beeldpunten, als beelden die als videogeheugen zijn opgeslagen, kunnen met deze optie worden opgeroepen.
- Het bewerken van beelden. Door dit programma-onderdeel wordt een ander programma geladen waarmee beeldbewerking kan worden uitgevoerd.
- Het printen van beelden op een printer of plotten op een plotter. Behalve een 'hard copy' op de printer kan met het printprogramma ook een beeld op schijf worden gezet (videogeheugen-dump). Zoals eerder opgemerkt, gaat het laden van een beeld voor gebruik in andere programma's sneller als het is opgeslagen als video-geheugeninhoud.

Zet verse koffie en vergaar de nodige moed voor de nu komende omvangrijke klus: het intypen van de programma's. Hier en daar kunnen groepjes regels worden gekopieerd. Let wel op de juiste regelnummering. Foutzoeken achteraf is vaak tijdrovender en lastiger dan simpel typewerk. Leg de ingetypte programma's op diskette vast onder de programmanaam zoals die in de eerste REM-regel van het desbetreffende programma staat.

```
10  REM ************ PIXEL ************
20  MAXFILES=2:KEY OFF
30  DIM C%(63,47)
40  OPEN"GRP:" FOR OUTPUT AS #1
50  SCREEN 2:COLOR 1,12,12:CLS
60  LINE(30,20)-(230,180),4,BF
70  LINE(10,5)-(80,40),10,BF
80  PSET(5,10),12
90  PRINT#1,"    M"
100 PSET(5,20),12
110 PRINT#1,"    S"
120 PSET(5,30),12
130 PRINT#1,"  PIXELS"
140 PSET(90,60),4:COLOR 15
150 PRINT#1,"HOOFDMENU"
160 PSET(90,80),5
170 PRINT#1,"l Opnemen"
180 PSET(90,90),4
190 PRINT#1,"2 Weergeven"
200 PSET(90,100),4
210 PRINT#1,"3 Bewerken"
220 PSET(90,110),4
230 PRINT#1,"4 Printen"
240 PSET(120,150),4
250 PRINT#1,"Kies nummer"
260 A$=INKEY$:IF A$="" THEN 260
270 A=VAL(A$)
280 IF A<1 OR A>4 THEN BEEP=GOTO 260
290 ON A GOTO 320,940,300,310
300 RUN "BEWERK"
310 RUN "UITVOER"
320 CLS
330 LINE(30,20)-(230,180),10,BF
340 LINE(10,5)-(80,40),9,BF
350 COLOR 1
360 PSET(5,10),12
370 PRINTtfl,"    M"
380 PSET(5,20),12
390 PRINT#1,"    S"
400 PSET(5,30),12
410 PRINT#1,"  PIXELS"
420 PSET(100,60),4
430 PRINT#1,"OPNEMEN"
440 PSET(40,80),10
450 PRINT#1,"* Afbeelding in printer"
460 PSET(80,135),10
470 PRINT#1," KIES KLEUR"
480 PSET(40,150),10
490 PRINT#1,"  l=Groen    2=Rood"
500 A$=INKEY$:IF A$="" THEN 500
510 A=VAL(A$)
520 IF A<1 OR A>2 THEN BEEP:GOTO 500
```

```
530 IF A=1 THEN C1%=12:C2%=2:C3%=3
ELSE C1%=6:C2%=8:C3%=9
540 LINE(40,60)-(230,180),10,BF
550 PSET(40,120),10
560 PRINT #1,"* Printer op zelftest"
570 PSET(40,150),10
580 PRINTttl,"* Druk op spatiebalk"
590 IF NOT STRIG(O) THEN 590
600 SCREEN 3:COLOR l,1,1:CLS
610 FOR Y=0 TO 47
620   IF NOT STRIG(l) THEN 620
630   FOR Q=l TO 100:NEXT Q
640   IF STRIG(l) THEN 640
650  FOR X=12 TO 50
660    K=STICK(1)
670    C%(X,Y)=C1%
680    IF K=3 THEN C%(X,Y)=C1%
690    IF K=5 THEN C%(X,Y)=C2%
700     IF K=7 THEN C%(X,Y)=C3%
710    PSET(4*X,4*Y),C%(X,Y)
720     FOR Q=l TO 2:NEXT Q:REM aan te passen
vertraging
730  NEXT X
740 NEXT Y
750 BEEP
760 COLOR 15
770 PSET(20,20),C1:PRINT#1,"opslaan"
780 PSET(0,100),C1:PRINT#1," (J/N)"
790 A$=INKEY$
800 IF A$="J" OR A$="j" THEN 820
810 IF A$="N" OR A$="n" THEN 50 ELSE 790
820 REM ********* PIXEL OPSLAG ***************
830 SCREEN 2:COLOR 1,10,10:CLS
840 GOSUB 1410
850 OPEN N$+".PIX" FOR OUTPUT AS #2
860 PRINT#2,CO%,C1%,C2%,C3%
870 FOR Y=0 TO 47
880   FOR X=0 TO 63
890    PRINT#2,C%(X,Y)
900  NEXT X
910 NEXT Y
920 CLOSE #2
930 SCREEN 0:COLOR 1,10,10:CLS
940 PRINT:"Plaats programmaschijf"
950 PRINT:"en druk op de spatiebalk"
960 IF NOT STRIG(0) THEN 960 970 GOTO 50
980 REM *********** WEERGEVEN ***************
990 SCREEN 2:COLOR 1,12,12:CLS
1000 LINE(30,20)-(230,180),9,BF
1010 LINE(10,5)-(80,40),13,BF
1020 PSET(5,10),12
1030 PRINT#1,"    M"
```

```
1040 PSET(5,20),12
1050 PRINT#l,"    S"
1060 PSET(5,30),12
1070 PRINTttl,"  PIXELS"
1080 PSET(90,60),9:COLOR l
1090 PRINT#l,"WEERGEVEN van BEELDEN"
1100 PSET(40,95),9
1110 PRINT#1,"! PIXELS  (.PIX)"
1120 PSET(40,115),9
1130 PRINT#1,"2 SCHERM  (.SCR)"
1140 PSET(120,160),9
1150 PRINT#1,"Kies nummer"
1160 A$=INKEY$:IF A$="" THEN 1160
1170 A=VAL(A$)
1180 IF A<1 OR A>2 THEN BEEP:GOTO 1160
1190 IF A=1 THEN T$=".PIX":GOSUB 1410
1200 IF A=1 THEN GOTO 1280
1210 IF A=2 THEN T$=".SCR":GOSUB 1410
1220 SCREEN 3:COLOR 1,1,1:CLS
1230 BLOAD N$+".SCR",S
1240 IF NOT STRIG(0) THEN 1240
1250 SCREEN 2:COLOR 1,12,12:CLS
1260 GOTO 50
1270 REM ******** BEELD OPROEPEN **********
1280 SCREEN 3:COLOR 1,1,1:CLS
1290 OPEN N$+".PIX" FOR INPUT AS #2
1300 INPUT #2,CO%,C1%,C2%,C3%
1310 FOR Y=0 TO 47
1320   FOR X=0 TO 63
1330     INPUT#2,C%(X,Y)
1340    PSET(4*X,4*Y),C%(X,Y)
1350   NEXT X
1360 NEXT Y
1370 CLOSE #2
1380 IF NOT STRIG(O) THEN 1380
1390 SCREEN 2:COLOR 1,12,12:CLS
1400 GOTO 50
1410 REM ************** SCHIJFINHOUD ************
1420 SCREEN 0:COLOR 1,10,10:CLS
1430 PRINT"    Plaats opslagschijf"
1440 PRINT:PRINT "    en druk op de spatiebalk"
1450 A$=INPUT$(1)
1460 CLS
1470 FILES
1480 INPUT "NAAM (zonder toevoeging)";N$
1490 CLS
1500 PRINT"Druk na beeldvorming op spatiebalk"
1510 FOR Q=1 TO 1000:NEXT Q
1520 SCREEN 2:COLOR 1,1,1:CLS
1530 RETURN
```

Dat was het hoofdprogramma. Door de verdeling in blokken is de loop van het programma goed te volgen. Als er vreemde dingen gebeuren, is er mogelijk ergens een %-teken weggevallen. De computer ziet C en C% als twee verschillende variabelen.
Het hoofdmenu wordt op het scherm gezet door het blok tussen de regels 10 en 290. Hier is dankbaar gebruik gemaakt van de mogelijkheid om ingekleurde blokken op het scherm te zetten met de LINE-opdracht. De schermopmaak bij een menu-gestuurd programma is belangrijk en door het gebruik van ondersteunende kleuren wordt een aantrekkelijk beeld gevormd. In de regels 300 en 310 treffen weverwijzingen aan naar onderdelen die in een apart programma zijn gezet. Ze worden door het hoofdprogramma aangeroepen, zodat de indeling van het totale beeldbewerkingsprogramma - in afzonderlijke programma's - niet storend is. Ten gerieve van de cassetterecorder-gebruik(st)ers zijn de programma's die de beeldbewerking verzorgen op zichzelf staand uitgevoerd. Als van die programma's de plaats op de cassetteband bekend is, kunnen ze dus afzonderlijk worden ingeladen en gebruikt. In de regels 320 tot en met 530 worden de voorbereidingen getroffen voor het opnemen van een beeld met de omgebouwde printer. De opname zelf wordt in het volgende blok gerealiseerd.
De regels 540 tot en met 810 vormen het software-deel van ons elektronicaproject (de beeldopnemer). Dit programmadeel houdt de stand van de eindschakelaar op de printer in de gaten en verwerkt de informatie van de sensor tot een beeld. Het is prachtig om het samenspel van dit programma-onderdeel en de printer te aanschouwen. Lijn voor lijn wordt de informatie vanaf het papier op het beeldscherm van de computer overgenomen.
Het nu verkregen beeld wordt in het massageheugen opgeslagen in de regels 820 tot en 930.
Het menu waaruit het beeldtype - dat we op het scherm willen zien - kan worden gekozen, is vastgelegd in de regels 940 tot en met 1230. Het oproepen van een beeld dat in beeldpunten is opgeslagen, duurt aanmerkelijk langer dan het oproepen van een beeld dat als inhoud van het videogeheugen is opgeslagen. Bovendien verschijnt het beeld bij gebruikmaking van de laatste methode in één keer op het scherm. Dit is zeer geschikt voor toepassingen in andere programma's (spelletjes, adventures enz.). In het nog volgende afdrukprogramma is de mogelijkheid ingebouwd om een beeld als videogeheugen-inhoud op schijf op te slaan.
Een subroutine sluit het programma af. Het gedeelte tussen de regels 1240 en 1370 verzorgt het oproepen van het beeld. De schijf-inhoud wordt weergegeven met behulp van de subroutine die op regel 1380 begint. Het nu volgende programma zorgt voor alle bewerkingen waarbij met kleuren en beeldpunten wordt gemanipuleerd. Het gedeelte waarmee plaatsveranderingen kunnen worden uitgevoerd, is als een afzonderlijk programma opgenomen.

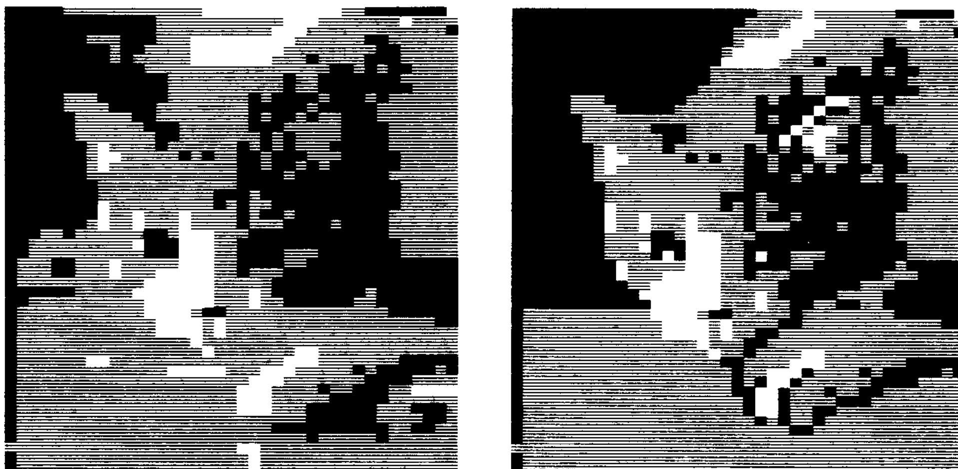
```
10 REM ************ BEWERK ****************
20 MAXFILES=2
30 DIM C%(63,47)
40 OPEN"GRP:" FOR OUTPUT AS #1
50 SCREEN 2:COLOR 1,10,10:CLS
60 PSET(20,0),4
70 PRINT#1,"GEBRUIK DE CURSORBESTURING"
80 LINE(10,20)-(115,60),4,BF
90 LINE(130,20)-(235,60),4,BF
100 COLOR 15
110 PSET(40,22),4:?#1,"KLEUR"
120 P3ET(20,35),4:?#1,"VERANDERING"
130 PSET(160,22),4:?#1,"PLAATS"
140 PSET(140,35),4:?#1,"VERANDERING"
150 Y=30:S=1
160 A=STICK(0)
170 IF A=7 THEN S=1
180 IF A=3 THEN S=2
190 COLOR 10:PSET(120,Y),10:PRINT#1, "•"
200 COLOR 1:PSET(120,Y),10:IF S=l THEN PRINT#1,"<"
 ELSE PRINT#1,">" 210 IF STRIG(0) THEN 230 220 GOTO 160
230 IF S=2 THEN PSET(136,130),10:PRINT#1,
 "WACHT !!!":RUN "GEO" 240 CLS:Y=130
250 LINE(45,120)-(200,160),4,BF
260 COLOR 15
270 PSET(50,130),4:PRINT#1,"Gehele beeld"
280 PSET(50,140),4:PRINT#1,"Individuele pixels"
290 A=STICK(0)
300 IF S=l AND A=5 THEN Y=140
310 IF S=l AND A=l THEN Y=130
320 COLOR 10:PSET(30,Y),10:PRINT#1,"B"
330 COLOR 1:PSET(30,Y),10:PRINT#1,"*"
340 IF NOT STRIG(O) THEN 290
350 IF S=1 AND Y=130 THEN GOTO 370
360 IF S=1 AND Y=140 THEN GOTO 1320
370 REM ************ KLEURVERANDERING ***********
380 CLS:PRINT"          KLEUREN"
390 LINE(20,0)-(180,170),4,BF
400 COLOR l:PSET(50,10),4:PRINT#1,">"
410 COLOR 15
420 PSET(30,10),4:PRINT#1," 0   Transparant"
430 PSET(30,20),4:PRINT#1"  1   Zwart"
440 PSET(30,30),4:PRINT#1, " 2   Groen"
450 PSET (30, 40), 4: PRINT#1," 3   Lichtgroen"
460 PSET(30,50),4:PRINT#1," 4   Blauw"
470 PSET(30,60),4:PRINT#1," 5   Lichtblauw"
480 PSET(30,70),4:PRINT#1," 6   Donkerrood"
490 PSET(30,80),4:PRINT#1," 7   Ciaan"
500 PSET(30,90),4:PRINT#1," 8   Rood"
510 PSET(30,100),4:PRINT#1," 9   Lichtrood"
520 PSET(30,110),4:PRINT#1, "10   Geel"
530 PSET(30,120),4:PRINT#1,"11   Lichtgeel"
```

```
540 PSET(30,130),4:PRINT#1, "12   Donkergroen"
550 PSET(30,140),4:PRINT#1, "13   Paars"
560 PSET(30,150),4:PRINT#1,"14   Wit"
570 PSET(30,160),4:PRINT#1, "15   Helder wit"
580 C=0
590 FOR K=0 TO 3
600 PSET(10,180),10:COLOR 10
605 PRINT#1 ,"••••••••••••••••••"
610 PSET(10,180),10:COLOR   l
615 PRINT#1,"Kies kleur ";K;"  (0=achtergrond)"
620 A=STICK(0)
630 FOR Q=l TO 20:NEXT Q
640 C=C-(A=5)+(A=1)
650 IF C<0 THEN C=0
660 IF C>15 THEN C=15
670 COLOR 4:PSET(50,C*10 + 10) ,4:PRINT#1, "•"
680 COLOR 1:PSET(50,C*10+10),4:PRINT#1,">"
690 IF STRIG(O) THEN 700 ELSE 620
700 F(K)=C
710 NEXT K
720 CO%=F(0):C1%=F(1):C2%=F(2):C3%=F(3)
730 A=l
740 GOSUB 1020
750 GOSUB 1140
760 FOR Y=0 TO 47
770 FOR X=0 TO 63
780 IF C%(X,Y)=0 THEN C%(X,Y)=F(0)
790 IF C%(X,Y)=C1 THEN C%(X,Y)=F(1)
800 IF C%(X,Y)=C2 THEN C%(X,Y)=F(2)
810 IF C%(X,Y)=C3 THEN C%(X,Y)=F(3)
820 PSET(4*X,4*Y),C%(X,Y)
830 NEXT X
840 NEXT Y
850 IF NOT STRIG(O) THEN 850
860 SCREEN O:COLOR 1,10,10:CLS
870 PRINT:PRINT:PRINT"   OPSLAAN (J/N) ";
880 A$=INPUT$(1)
890 IF A$="N" OR A$="n" THEN GOTO 1620
900 IF A$="J" OR A$="j" THEN GOSUB 1020 ELSE 880
910 REM ********* OPSLAG OP SCHIJF **************
920 CLS:PRINT"OPSLAAN van ";N$;".PIX"
930 OPEN N$+".PIX" FOR OUTPUT AS #2
940 PRINT#2,CO%,C1%,C2%,C3%
950 FOR Y=0 TO 47
960 FOR X=0 TO 63
970 PRINT#2,C%(X,Y)
980 NEXT X
990 NEXT Y
1000 CLOSE #2
1010 GOTO 1620
```

```
1020 REM ************* SCHIJFINHOUD *************
1030 SCREEN 0:COLOR 1,10,10:CLS
1040 PRINT"   Plaats opslagschijf"
1050 PRINT"   en druk op de spatiebalk"
1060 FOR Q=1 TO 50:NEXT Q
1070 IF NOT STRIG(0) THEN 1070
1080 CLS
1090 FILES:PRINT
1100 PRINT"Alleen sterren invullen"
1110 PRINT"Naam  ********.PIX"
1120 INPUT"    ";N$
1130 RETURN
1140 REM ********** BEELD OPROEPEN *************** 1150 CLS
1160 PRINTDRUK NA BEELDVORMING OP SPATIEBALK" 1170 FOR Q=l TO
1000:NEXT Q 1180 SCREEN 3:COLOR 1,1,1 1190 CLS
1200 OPEN N$+".PIX" FOR INPUT AS #2 1210 INPUT#2,C0%,C1%,C2%)C3%
1220 FOR Y=0 TO 47 1230  FOR X=0 TO 63 1240     INPUT#1,C%(X,Y)
1250      PSET(4*X,4*Y),C%(X,Y) 1260    NEXT X 1270 NEXT Y 1280
PRINT 1290 CLOSE #2
1300 IF NOT STRIG(O) THEN 1300 1310 RETURN
1320 REM *********** PIXEL VERANDERING ***********
1330 GOSUB 1020
1340 GOSUB 1140
1350 XO=0:YO=0:Y=0
1360 A=STICK(0)
1370 X=XO-(A=2)-(A=3)-(A=4)+(A=6)+(A=7)+(A=8)
1380 IF X<0 THEN X=0
1390 IF X>63 THEN X=63
1400 Y=YO+(A=1)+(A=2)+(A=8)-(A=4)-(A=5)-(A=6)
1410 IF Y<0 THEN Y=0
1420 IF Y>47 THEN Y=47
1430 PSET(4*X,4*Y),15
1440 FOR Q=1 TO 10:NEXT Q
1450 PSET(4*X,4*Y),1
1460 FOR Q=1 TO 10:NEXT Q
1470 IF STRIG(0) THEN GOSUB 1480
1480 IF X=XO AND Y=YO THEN 1360
1490 PSET(4*XO,4*YO),C%(XO,YO)
1500 XO=X:YO=Y
1510 GOTO 1360
```

```
1520 REM *********** KLEURVERANDERING *************
1530 A=STICK(0)
1540 IF A=1 THEN GOSUB 1020
1550 IF A=1 THEN GOTO 910
1560 T=T+(A=7)-(A=3)
1570 IF T<1 THEN T-1
1580 IF T>15 THEN T=15
1590 C%(X,Y)=T
1600 PSET(4*X,4*Y),C%(X,Y)
1610 IF STRIG(0) THEN RETURN ELSE 1530
1620 REM ********* TERUG NAAR HOOFDPROGRAMMA ***********
1630 CLS
1640 PRINT "Plaats programmaschijf"
1650 PRINT "en druk op de spatiebalk"
1660 IF NOT STRIG(0) THEN 1620
1670 RUN "PIXEL"
```

Zoals we intussen gewend zijn, is ook hier weer een blokstructuur te herkennen. Het programma begint met een menu waaruit kan worden gekozen voor plaats- of kleurveranderingen (regels 10-200). Regel 230 verwijst naar een programma-onderdeel dat nog volgt.
Als we voor een kleurverandering kiezen, komen we in een submenu terecht waaruit kan worden gekozen voor de bewerking van het gehele beeld, of voor de bewerking van afzonderlijke beeldpunten. De selecties uit de menu's worden met behulp van de cursorbesturingstoetsen gemaakt.
De subroutines voor het weergeven van de schijfinhoud en het opslaan van de beelden kennen we al uit het vorige programma.
Op dit punt aangekomen, kunnen we beelden opnemen en weergeven. Bovendien kunnen de beelden worden gemanipuleerd. Wellicht is het een welkome onderbreking van het typewerk om eens een plaatje in het geheugen op te



Afbeelding 30-1 Een kat (links) en een geretoucheerde kat (rechts).

slaan en er vervolgens mee aan de slag te gaan. Vooral het bewerken van afzonderlijke pixels (beeldpunten) biedt interessante mogelijkheden. De plaatjes in afbeelding 30-1 laten hier iets van zien.
Het eerste beeld is weergegeven, precies zoals het is opgenomen. Het is opmerkelijk hoe weinig informatie de 'menselijke computer' (ook wel hersenen genoemd) nodig heeft om details te herkennen. Bekijk het prentje maar eens door uw oogharen vanaf enige afstand. De beeldpunten vloeien in elkaar over en het oorspronkelijke beeld wordt weer herkenbaar. Het tweede beeld is een bewerking van het eerste. Zo zijn in de ogen pretlichtjes aangebracht en zijn de linkervoorpoot en het linkeroor (voor de kijkers rechts) wat meer geaccentueerd.
De afbeelding wordt tijdens de opname min of meer centraal op het beeldscherm gezet. Die plaats is afhankelijk van de snelheid van de printer en de snelheid waarmee de beeldpunten op het beeldscherm van de computer worden gezet.
Het volgende programma maakt het mogelijk het beeld over het beeldscherm te schuiven, het beeld te spiegelen, te verdubbelen enz. Dit nieuwe beeld kan worden opgeslagen op schijf of cassette. De programmanaam 'GEO' slaat op de geometrische bewerkingen die het programma kan uitvoeren.

```
10 REM *******GEO********************
20 KEY OFF:MAXFILES=2
30 DIM C%(63,47)
40 OPEN"GRP:" FOR OUTPUT AS #1
50 SCREEN 2:COLOR 1,12,12:CLS
60 LINE(130,20)-(235,192),4,BF
70 COLOR 15
80 PSET(160,22), 4 : PRINT#1,"Plaats"
90 PSET(140,35),4:PRINT#1,"verandering"
100 PSET(136,90),4:PRINT#1,"Hor/Vert"
110 PSET(136,100),4:PRINT#1,"Verdubbelen"
120 PSET(136,100),4:PRINT#1,"Omkeren"
130 PSET(136,120),4:PRINT#1,"Spiegelen"
140 PSET(136,130),4:PRINT#1,"Waterspiegel"
150 PSET(136,140),4:PRINT#1,"Speelkaart"
160 Y=90
170 A=STICK(0)
180 IF A=5 THEN Y=Y+10
190 IF Y>140 THEN Y=140
200 IF A=l THEN Y=Y-10
210 IF Y<90 THEN Y=90
220 PSET(120,Y),12:COLOR 12:PRINT#1," • "
230 PSET(120,Y),12:COLOR l:PRINT#1,"*"
240 IF STRIG(0) THEN 260
250 GOTO 170
260 IF Y=90 THEN N=l
270 IF Y=100 THEN N=2
280 IF Y=110 THEN N=3
290 IF Y=120 THEN N=4
300 IF Y=130 THEN N=5
```

```
310 IF Y=140 THEN N=6
320 SCREEN 0:COLOR 1,10,10:CLS
330 ON N GOTO 340,650,790,910,1040,1160
340 PRINT"Verschuiving (links= -)"
350 PRINT"             (rechts=+)"
360 INPUT"Aantal beeldpunten ";N
370 CLS
380 PRINT"Verschuiving (Omhoog=-)"
390 PRINT"             (Omlaag=+)"
400 INPUT"Aantal beeldpunten ";M
410 GOSUB 1280
420 GOSUB 1540
430 IF N<0 THEN A=0:B=63:ST=1 ELSE A=63:B=0:ST=-1
440 IF M<0 THEN D=0:E = 47:EP=1 ELSE D=47:E=0:EP=-1
450 FOR Y=D TO E STEP EP
460  FOR X=A TO B STEP ST
470   XN=X+N:YN-Y+M
480   PSET(4*XN,4*YN),C%(X,Y)
490   IF XN<0 THEN XN=0
500   IF XN>63 THEN XN=63
510   IF YN<0 THEN YN=0
520   IF YN>47 THEN YN-47
530   C%(XN,YN)=C%(X,Y)
540  NEXT X
550 NEXT Y
560 IF M<0 THEN D=48+M:E=47 ELSE D=0:E=M-1
570 FOR Y=D TO E
580  FOR X=0 TO 63
590   PSET(4*X,4*Y),1
600  NEXT X
610 NEXT Y
620 IF NOT STRIG(0) THEN 620
630 GOSUB 1280
640 GOTO 1390
650 REM ************ VERDUBBELEN *************
660 GOSUB 1280
670 GOSUB 1540
680 FOR Y=0 TO 47
690  FOR X=0 TO 32
700   PSET(4*X,4*Y),C%(X,Y)
710   PSET(4*(X+31),4*Y),C%(X,Y)
720   C%(X+31,Y)=C%(X,Y)
730  NEXT X
740 NEXT Y
750 IF NOT STRIG(0) THEN 750
760 GOSUB 1280
770 GOTO 1390
790 REM ********** OMKEREN *****************
800 GOSUB 1280
810 GOSUB 1540
```

```
820 FOR Y=0 TO 47
830 FOR X=0 TO 63
840 PSET(4*(63-X),4*Y),C%(X,Y)
850 C%(63-X,Y)=C%(X,Y)
860 NEXT X
870 NEXT Y
880 IF NOT STRIG(O) THEN 880
890 GOSUB 1280
900 GOTO 1390

910 REM ************ SPIEGELEN **************
920 GOSUB 1280
930 GOSUB 1490
940 FOR Y=0 TO 47
950 FOR X=0 TO 32
960 PSET(4*X,4*Y),C%(X,Y)
970 PSET(4*(62-X),4*Y),C%(X,Y)
980 C%(62-X,Y)=C%(X,Y)
990 NEXT X
1000 NEXT Y
1010 IF NOT STRIG(O) THEN 1010
1020 GOSÜB 1280
1030 GOTO 1390

1040 REM **************WATERSPIEGEL ************
1050 GOSUB 1280
1060 GOSUB 1540
1070 FOR Y=0 TO 23
1080 FOR X=0 TO 63
1090 PSET(4*X,4*(47-Y)),C%(X,Y)
1100 C%(X,47-Y)=C%(X,Y)
1110 NEXT X
1120 NEXT Y
1030 IF NOT STRIG(0) THEN 1030
1140 GOSUB 1280
1150 GOTO 1390

1160 REM ************** SPEELKAART **************
1170 GOSUB 1280
1180 GOSUB 1540
1190 FOR Y=0 TO 23
1200 FOR X=0 TO 63
1210 PSET(4*(63-X),4*(47-Y)),C%(X,Y)
1220 C%(63-X,47-Y)=C%(X,Y)
1230 NEXT X
1240 NEXT Y
1250 IF NOT STRIG(0) THEN 1250
1260 GOSUB 1280
1270 GOTO 1390
1280 REM ******** BESTANDEN LEZEN *************
1290 SCREEN 0:COLOR 1,10,10:CLS
```

```
1300 PRINT "Plaats opslagschijf"
1310 PRINT "en druk op de spatiebalk"
1320 IF NOT STRIG(O) THEN 1320
1330 CLS
1340 FILES
1350 PRINT"Alleen sterren invullen"
1360 PRINT"Naam  ********.PIX"
1370 INPUT"     ";N$
1380 RETURN
1390 REM ********** BEELDOPSLAG *****************
1400 CLS:PRINT "OPSLAAN ";N$;".PIX"
1410 OPEN N$+".PIX" FOR OUTPUT AS #2
1420 PRINT#2,CO%,C1%,C2%,C3%
1430 FOR Y=0 TO 47
1440   FOR X=0 TO 63
1450    PRINT#2,C%(X,Y)
1460   NEXT X
1470 NEXT Y
1480 CLOSE #2
1490 SCREEN 0:COLOR 1,10,10:CLS
1500 PRINT "Plaats programmaschijf"
1510 PRINT "en druk op spatiebalk"
1520 IF NOT STRIG(0) THEN 1520
1530 RUN "PIXEL"
1540 REM ******** BEELD INLEZEN *****************
1550 CLS:PRINT"Druk NA BEELDVORMING op spatiebalk"
1560 FOR Q=1 TO 1000:NEXT Q
1570 SCREEN 3:COLOR 1,1,1:CLS
1580 OPEN N$+".PIX" FOR INPUT AS #2
1590 INPUT#2,CO%,C1%,C2%,C3%
1600 FOR Y=0 TO 47
1610   FOR X=0 TO 63
1620     INPUT#2,C%(X,Y)
1630     PSET(4*X,4*Y),C%(X,Y)
1640   NEXT X
1650 NEXT Y
1660 CLOSE #2
1670 IF NOT STRIG(0) THEN 1670
1680 RETURN
```

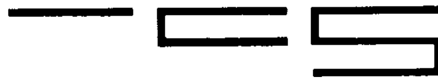
Ongetwijfeld herkent u de reeds behandelde beeldbewerkingen, zoals spiegelen en omkeren. Ook wat het opslaan en lezen van de beelden betreft, is er weinig nieuws onder de zon.
De drie grootste programma's zijn nu klaar. Afbeeldingen kunnen in alle mogelijke kleuren en standen op het scherm worden gezet. Het enige dat nog ontbreekt is het maken van een 'hard copy'. Zoals afgesproken, maken we het afdrukprogramma geschikt voor:
- het maken van afdrukken op een printer,
- het maken van afdrukken op een plotter en
- het 'dumpen' van de video-geheugeninhoud op diskette.

```
10 REM ***********UITVOER***************
20 MAXFILES=2
30 DIM C%(63,47)
40 OPEN"GRP:" FOR OUTPUT AS #1
50 SCREEN 2:COLOR 1,12,12:CLS:KEY OFF
60 LINE(30,20)-(230,180),4,BF
70 LINE(10,5)-(80,40),10)BF
80 PSET(5,10),12
90 PRINT#1,"    M"
100 PSET(5,20),12
110 PRINT#1,"    S"
120 PSET(5,30),12
130 PRINT#1,"  PIXELS"
140 PSET(90,60),4:COLOR 15
150 PRINT#1,"UITVOER"
160 PSET(90,80),4:PRINT#1, "l Dump op schijf"
170 PSET(90,95),4:PRINT#1,"2 Printer"
180 PSET(90,110),4:PRINT#1, "3 Plotter-
190 PSET(120,150),4:PRINT#1,"Kies nummer"
200 A$=INKEY$:IF A$="" THEN 200
210 A-VAL(A$)
220 IF A<1 OR A>3 THEN BEEP:GOTO 200
230 ON A GOTO 240,390,440
240 REM ******3,5 INCH-SCHIJF********************
250 SCREEN 0:COLOR 1,10,10:CLS
260 PRINT "Plaats de opslagschijf voor"
280 PRINT "het beeld pas NADAT het beeld"
300 PRINT "volledig gevormd is"
310 PRINT
320 PRINT"DRUK DAARNA OP DE SPATIEBALK"
330 FOR Q=1 TO 2000:NEXT Q
340 GOSUB 740
350 GOSUB 830
360 IF NOT STRIG(0) THEN 360
370 BSAVE N$+".SCR",0,16383,S
380 GOTO 1030
390 REM ************ PRINTER ****************
400 GOSUB 740
410 GOSUB 830
420 PR=1
430 GOTO 1030
440 REM *********** PLOTTER *****************
450 GOSUB 740
460 SCREEN 0:COLOR 1,10,10:CLS
470 PRINT"Beeldhoogte 3 of 6 centimeter";
480 A$=INPUT$(1)
490 IF A=3 THEN A=1:GOTO 520
500 IF A=6 THEN A=2:GOTO 520
510 BEEP:GOTO 460
520 CLS:PRINT'PLOTTER UITVOER"
530 OPEN N$+".PIX" FOR INPUT AS #2
```

```
540 INPUT#2,CO%,C1%,C2%,C3%
550 LPRINT N$
560 LPRINT CHR$(&HlB)+"#"
570 FOR Y=0 TO 47
580 FOR R=1 TO A
590 FOR X=0 TO 63
600 IF R=2 THEN 620
610 INPUT#2,C%(X,Y)
620 FOR P=l TO A
630 IF C%(X,Y)=C1% THEN LPRINT "J4,0,0,1,-
 4,0,0,1,4,0":LPRINT"RO,-2"     640     IF     C%(X,Y)=C2%     THEN
LPRINT"J4,O,O,l,-
 4,O":LPRINT"R4,-l"
650 IF C%(X,Y)=C3% THEN LPRINT"J4,0"
660 IF C%(X,Y)-CO% THEN LPRINT"R4,0"
670 NEXT P
680 NEXT X
690 LPRINT"F":LPRINT"RO,17"
700 NEXT R
710 NEXT Y
720 CLOSE #2
730 GOTO 1030
740 REM ********* BESTANDEN LEZEN *************
750 SCREEN O:COLOR 1,10,10:CLS
760 PRINT "Plaats opslagschijf"
770 PRINT "en druk op de spatiebalk"
780 IF NOT STRIG(0) THEN 780
790 CLS
800 FILES
810 INPUT "NAAM ";N$
820 RETURN
830 REM ************* BEELD UITLEZEN *********
840 SCREEN 3:COLOR 1,1,1:CLS
850 OPEN N$+".PIX" FOR INPUT AS #2
860 INPUT#2,CO%,C1%,C2%,C3%
870 FOR Y=0 TO 47
880 FOR X=0 TO 63
890 INPUT#2,C%(X,Y)
900 IF NOT PR THEN 950
910 IF C%(X,Y)=CO% THEN P$=" "
920 IF C%(X,Y)=C1% THEN P$="#"
930 IF C%(X,Y)=C2% THEN P$="="
940 IF C%(X,Y)=C3% THEN P$="-"
950 LPRINT P$;
960 PSET(4*X,4*Y),C%(X,Y)
970 NEXT X
980 IF NOT PR THEN 1000
990 LPRINT
1000 NEXT Y
1010 CLOSE #2
1020 RETURN
1030 SCREEN 0:COLOR 1,10,10: CLS
```

```
1040 PRINT:"Plaats programmaschijf"
1050 PRINT "en druk op spatiebalk"
1060 IF NOT STRIG(0) THEN 1060
1070 RUN "PIXEL"
```

Het is u misschien opgevallen dat er in het programmablok PRINTER niets wordt geprint. Dat gebeurt in werkelijkheid tegelijk met het inlezen van het beeld. Het lijkt inconsequent, maar deze opzet bespaart wat werk. Als u veranderingen in de programma's wilt aanbrengen, bent u daar natuurlijk vrij in. Omdat het plotterdeel er wat eigenaardig uitziet, kunnen er gemakkelijktikfou-tjes insluipen. Afbeelding 30-2 laat zien hoe de plotter dunne en dikke lijntjes maakt.



Afbeelding 30-2 Plotpatronen.

Een plotterafbeelding is drie centimeter hoog. Door het fijne lijnenpatroon blijven de details goed zichtbaar. Misschien is zo'n plotterafbeelding nog wel mooier dan de afbeelding op het beeldscherm... Willen we een afbeelding tweemaal zo groot maken (zes centimeter hoog), dan laten we de plotterpatronen tweemaal naast elkaar en iedere regel tweemaal onder elkaar afdrukken. (Op deze manier worden de blokjes waaruit de afbeelding bestaat, vier keer zo groot.) Op A4-papier kan de afbeelding eventueel nog groter worden getekend. Geef de variabele A in regel 620 de waarde 3 (of 4). De tijd die nodig is om het beeld te plotten, wordt hierdoor wel aanzienlijk vergroot. Voor een MSX-printer worden met de tekens

= [SHIFT][GRAPH] =

= is gelijk-teken

\- minteken

de beste resultaten verkregen.

Voor andere printers zou u de tekens #, = en - kunnen gebruiken. Het beste kunt u wat experimenteren met verschillende tekens. Het mooiste resultaat verkrijgt u door zelf karakters te definiëren. Deze worden als bit-patronen naar de printer verzonden. Raadpleeg de handleiding van uw printer hierover. Een juiste instelling van de pitch en de regelafstand garanderen een (bijna) naadloze aaneensluiting van de tekens.

Een andere manier om een 'hard copy' van het beeldscherm te maken, is het fotograferen van het beeldscherm. De beeldschermen laten zich goed vastleggen op dia's of foto's. Door het enigszins gebolde beeldscherm kan er gemakkelijk randvertekening ontstaan, vooral als we het scherm vanaf een kleine afstand fotograferen. Door een lens met een wat langer brandpunt te nemen, wordt de afstand tot het scherm al gauw groter, waardoor de vertekening verdwijnt. Let er wel op dat het beeldscherm goed schoon is: op een foto vallen allerlei vlekjes, vuiltjes en vingerafdrukken extra op.

Het TV-beeld wordt 50 keer per seconde herhaald. Oudere fototoestellen hebben nog een sluitertijd van 1/25 seconde. In die tijd wordt het beeld precies tweemaal geschreven. De modernere camera's hebben een aan de internationale normen aangepaste sluitertijd van 1/30 seconde. Als met deze sluitertijd een opname wordt gemaakt, zien we een hinderlijke streep, omdat de beeldvorming nog niet compleet is. Beter is het dan ook om een lange sluitertijd te nemen. In dat geval nemen we tientallen keren hetzelfde beeld op en is het aandeel van een nog niet voltooid beeld te verwaarlozen. De sluitertijd die het beste voldoet, is een halve seconde. Dit houdt wel in dat het fototoestel op een statief moet staan. Het diafragma zal een instelling tussen 8 en 22 moeten hebben, afhankelijk van de omstandigheden. Probeer de opname beeldvullend te maken, zodat de schermranden niet zichtbaar zijn (die kunt u overigens ook later van de foto afknippen). Door de film tweemaal te belichten - eenmaal met een titel en eenmaal met een gedigitaliseerd beeld - kunnen prachtige aankondigingen in diaseries worden gemaakt. Ook voor zelfgemaakte videofilms zijn opgeslagen computerbeelden uitermate geschikt.

31 Beeldmanipulatie met de MSX2-computer

Als we denken aan grafische computertoepassingen, kunnen we niet om de MSX2-computer heen. In dit hoofdstuk zullen we even proeven aan de mogelijkheden die deze fantastische machine biedt. Het is niet moeilijk om de door ons opgenomen plaatjes op het hogeresolutiescherm van de MSX-computer af te beelden, maar wel op dat van de MSX2-computer. Het programma PIXEL is gemakkelijk aan te passen: vervang alle SCREEN 3-opdrachten in SCREEN 2-opdrachten. In de PSET-opdrachten moeten de factoren 4 worden verwijderd, anders komen de afgebeelde puntjes te ver uit elkaar te liggen. Dus niet PSET(4*X,4*Y),C%(X,Y) maar PSET(X,Y),C%(X,Y).
Het nu verkregen beeld is maar 63 bij 47 beeldpunten groot, pasfotoformaat. Het is echter wel van zeer goede kwaliteit. Zo'n pasfoto doet het goed in andere programma's. Met een MSX2-computer kunnen we zo'n scherp prentje op iedere plaats op het hogeresolutiescherm zetten door bij de coördinaten van de beeldpunten, de coördinaten van de gewenste linkerbovenhoek op te tellen. De opdracht PSET(X+100,Y+70),C%(X,Y) zet het plaatje ongeveer in het midden van het beeldscherm.
Het inlezen van een beeld gaat relatief langzaam. Het is met een MSX-computer niet zo aantrekkelijk om de beelden over het scherm te laten schuiven. Met de MSX2-computer daarentegen wel. Deze machine biedt unieke grafische mogelijkheden die in veel opzichten vergelijkbaar zijn met de fraaie staaltjes van grafische effecten die omroeporganisaties gebruiken in hun aankondigingen.
MSX2-BASIC heeft extra beeldbewerkingsopdrachten, zoals SET PAGE en COPY.
Het volgende programma is ontleend aan het eerder behandelde programma PIXEL. Het is echter sterk ingekort en niet in de tot nu toe opgebouwde programmacluster opgenomen. Zie het als een demonstratieprogramma voor de MSX2-computerbezit(s)ters. Het beeld wordt weergegeven op één van de hogeresolutieschermen. In het programma is SCREEN 5 gebruikt.

```
10     REM MSX2
20     MAXFILES=2:KEY OFF
30     DIM  C%(63,47)
40  SCREEN5:SET PAGE 1,1:COLOR 1,10,10:CLS
50  OPEN"GRP: "FOR OUTPUT AS #1
60  T$="PIX":GOSUB 500
70  SCREEN 5:COLOR 1,1,1:CLS
80  SET PAGE 0,0
90  OPEN"A:"+N$+".PIX" FOR INPUT AS #2
100 INPUT #2,CO%,C1%,C2%,C3%
110 FOR Y=0 TO 47
120   FOR X=0 TO 63
130    INPUT#2,C%(X,Y)
140    PSET(X,Y),C%(X,Y)
150   NEXT X
160 NEXT Y
```

```
170 CLOSE #2
180 REM manipulatie
190 COPY (0,0)-(63)47),0 TO (0,0),1
200 FOR X=0 TO 3
210  FOR Y=0 TO 3
220    COPY (0,0)-(63,47) TO (X*63,Y*47)
230     FOR Q=1 TO 200:NEXT Q
240  NEXT Y
250 NEXT X
260 FOR Q=1 TO 2500:NEXTQ
270 SET PAGE 1,1
280 FOR I=1 TO 4
290   COPY (0,0)-(63,47),0 TO (40*I,34*I)
300   FOR Q=1 TO 100:NEXT Q
310 NEXT I
320 FOR Q=1 TO 2000:NEXT Q
330 FOR N=1 TO 5
340   SET PAGE 0,0
350   FOR Q=1 TO 500:NEXT Q
360   SET PAGE 1,1
370   FOR Q=1 TO 500:NEXT Q
380 NEXT N
390 FOR I=1 TO 200
400   COPY(0,0)-(63,47),0 TO (I,0)
410 NEXT I
420 FOR I=1 TO 200
430   COPY(0,48)-(63,96),0 TO (I,48)
440 NEXT I
450 FOR I=1 TO 200
460   COPY(0,97)-(63,212),0 TO (I,97)
470 NEXT I
480 SET PAGE 0,0
490 IF STRIG(0) THEN END ELSE GOTO 490
500 REM laat schijfinhoud zien
510 SCREEN0:COLOR 1,10,10:CLS
520 PRINT'Tlaats opslagdiskette..."
530 PRINT:PRINT:PRINT"en druk op spatiebalk ";
540 A$=INPUT$(1)
550 CLS
560 FILES
570 PRINT"Alleen sterren invullen"
580 PRINT"Naam  ********.";T$
590 INPUT"    ";N$
600 CLS
610 SCREEN2:COLOR 1,1,1:CLS
620 RETURN
```

In de programmaregels 10 tot en met 170 wordt het beeld in de linkerbovenhoek van het beeldscherm geplaatst. Vanuit de subroutine aan het einde van het programma, maakt de gebruik(st)er een keuze uit de af te beelden plaatjes. Let even op regel 40. Hier wordt kennelijk nogal overbodig SCREEN 5 aangeroepen. Toch is deze regel belangrijk. In SCREEN-modus 5 kunnen we vier onafhankelijke schermbeelden opslaan en bewerken. Deze beelden staan op pagina (PAGE) O tot en met 3. Als de SCREEN 5-opdracht wordt uitgevoerd, wordt alleen de eerste pagina gewist. Alle andere pagina's kunnen dus nog beelden bevatten uit eerder uitgevoerde programma's. Omdat we in dit voorbeeld twee pagina's gebruiken, wordt in regel 40 pagina 1 voor alle zekerheid gewist. Elke nieuwe programma-run begint met een schone lei. Bovendien geven we de achtergrond alvast een kleurtje.

```
SCREEN  5:SET   PAGE   l,l:COLOR   l,10,10:CLS
```

Het eerste cijfer in de SET PAGE-opdracht geeft aan welke pagina op het scherm wordt afgebeeld. Het tweede cijfer bepaalt welke pagina in bewerking is. Het is mogelijk om een plaatje af te beelden, terwijl intussen veranderingen aan een ander worden aangebracht. Dit plaatje staat in het videogeheugen. SET PAGE 0,0 wil zeggen dat beeldpagina O op het scherm staat en dat bovendien alle beeldmanipulaties op dit beeld betrekking hebben. SET PAGE 1,0 wil zeggen dat we pagina 1 zien en dat de beeldbewerkingen plaatsvinden op de onzichtbare pagina 0.
Behalve de SET PAGE-opdracht speelt ook de COPY-opdracht een belangrijke rol in dit programma. In regel 190 wordt een deel van beeldpagina O naar een positie op de nog onzichtbare pagina 1 gekopieerd. Als we achter de coördinaten van de COPY-opdracht aangeven om welke pagina het gaat, is het niet belangrijk welke pagina op dat ogenblik op het scherm staat. Als we naar het huidige beeldscherm willen kopiëren, hoeven we geen paginacijfer op te geven.
We zullen de loop van het programma eens nagaan. De regels 180 tot en met 250 bevatten twee lussen. Eerst wordt het beeld viermaal gekopieerd. Telkens verandert de Y-coördinaat: er komen vier beeldjes onder elkaar te staan. Vervolgens wordt de X-coördinaat verhoogd en worden opnieuw vier beeldjes onder elkaar afgebeeld. Dit gaat zo door totdat er vier rijtjes van vier beelden op het scherm staan. Omdat er geen paginanummer achter de COPY-opdracht staat, gebeurt dit alles op de zichtbare pagina (PAGE 0). Als we regel 230 weglaten, gebeurt het kopiëren zó snel dat het voor ons oog niet meer bij te houden is: het lijkt alsof alle zestien plaatjes in één keer op het scherm worden gezet. Door de vertraging in regel 230 ontstaat het effect alsof de prentjes als kaarten over een tafel worden uitgespreid.
Regel 270 maakt PAGE 1 zichtbaar. We hadden er al een beeldje ingezet toen de pagina nog niet zichtbaar was (regel 90). Nu plaatsen we er op de diagonaal van het beeldscherm nog wat afbeeldingen bij (regel 280-310). Eenmaal opgeslagen beelden kunnen met een ongekende snelheid op het beeldscherm worden gezet. De programmaregels 330 tot en met 380 laten pa-

gina 0 en pagina 1 vijfmaal afwisselend op het scherm verschijnen. Hoe waanzinnig snel dit kan, kunt u zien als u de vertragingen uit de regels 350 en 370 verwijdert. Laat het beeld 100 keer wisselen.
Animatie met onze grafische juweeltjes behoort ook tot de MSX2-mogelijkheden. We kunnen de beeldjes als sprites over het scherm laten bewegen. Het laatste deel van het programma laat hier enkele voorbeelden van zien. De snelheid waarmee de MSX2-computer met gecompliceerde beelden omgaat, is verbazingwekkend. Dat er nog veel meer mogelijkheden in deze machine schuilen, bewijzen twee opdrachten (die niet in alle handboeken zijn opgenomen). MSX2-BASIC kent de opdracht COPY SCREEN [<mode>] waarmee externe videobeelden gedigitaliseerd in het geheugen kunnen worden opgeslagen. De opdracht SET VIDEO <mode> [,<ym>[,<cb>[,<sync> [,<audio>[,<video input>[,<AV control>]]]]]] laat zien dat er rekening is gehouden met koppeling aan video-apparatuur. Met SET VIDEO kunnen verschillende beelden worden 'gemixed' (superimpose). De verschillende parameters maken een flexibele aanpassing mogelijk:

<mode>	0 tot 3	(waar komen de beelden vandaan)
<ym>	0 of 1	(beeldintensiteit half of vol)
<cb>	0 of 1	(kleurenbesturing voor invoer of uitvoer)
<sync>	0 of 1	(interne of externe synchronisatie)
<audio>	0 tot 3	(regeling audiosignaal)
<video>	0 of 1	(invoer via RGB-of TV-aansluiting)
<AV>	0 of 1	(audio/video-output via RGB-aansluiting)

Over het concrete gebruik van de computer/video-koppeling is nog weinig bekend. Het programma in dit hoofdstuk heeft echter laten zien dat er nog steeds een uitdagend terrein van grafische mogelijkheden voor ons open ligt.

32 Tot slot

In dit gedeelte van het boek heeft u kennis kunnen maken met de intrigerende wereld van beeldbewerking. We hebben hierbij gebruik gemaakt van het lage-resolutiescherm. Voor gecompliceerde taken op het hogeresolutiescherm leent de MSX-computer zich minder goed; de MSX2-computer daarentegen wel. Echter, als u rekening houdt met de regel dat op het hogeresolutiescherm telkens acht beeldpunten naast elkaar dezelfde kleur moeten hebben, kunt u ook op een MSX-computer van het hogeresolutiescherm gebruikmaken en de programma's daaraan aanpassen.
Het oplossend vermogen van de in dit boek beschreven beeldopnemer kan drastisch worden verhoogd door gebruik te maken van lenzen. Als u die wilt toepassen, kunt u ook het beste gebruik maken van een gescheiden zend- en ontvangdiode. Als u geen financiële beperkingen kent, is de toepassing van een opnemer uit een streepjescodelezer ideaal. De printer moet dan wel in staat zijn om kleine regelverschuivingen uit te voeren. Ondanks de mogelijke verbeteringen is er geen reden om ontevreden te zijn met het behaalde resultaat. Het brengt beeldbewerkingstechnieken binnen de mogelijkheden van de homecomputer-bezit(s)ter en geeft daarmee een nieuwe dimensie aan haar of zijn werkzaamheden met dit fascinerende apparaat.